The Complexity of Two-Person Zero-Sum Games in Extensive Form

Daphne Koller* Nimrod Megiddo†

November 1990

This paper investigates the complexity of finding max-min strategies for finite two-person zero-sum games in the extensive form. The problem of determining whether a player with imperfect recall can guarantee himself a certain payoff is shown to be NP-hard. When both players have imperfect recall, this is even harder. Moreover, the max-min behavior strategy of such a player may use irrational numbers. Thus, for games with imperfect recall, computing the max-min strategy or the value of the game is a hard problem. For a game with perfect recall, we present an algorithm for computing a max-min behavior strategy, which runs in time polynomial *in the size of the game tree.*

1980 Mathematical Subject Classification numbers: 90D05, 68Q99.

1. Introduction

Until recently, little attention was devoted to computational aspects of solution concepts in game theory. The computational complexity of different solution concepts can vary drastically. Some work has been done on analyzing the complexity of computing a Nash Equilibrium in general normal form games, but this is still an open problem. Variations of this problem are known to be NP-hard (see Gilboa and Zemel [1989]). Complexity analysis of other solution concepts was done in Deng and Papadimitriou [1989]. Also, see Lucas [1972] for a survey. In this paper, we examine the complexity of solving two-person zero-sum games in extensive form.

In game theory (see, for example, Owen [1982], or McKinsey [1952]) there are two main forms for representing games, the *normal form*, giving the payoffs for all combinations of strategy choices, and the *extensive form*, describing the game as a tree, which

*Department of Computer Science, Stanford University, Stanford, California 94305, and IBM Almaden Research Center, 650 Harry Road, San Jose, California 95120-6099.

†IBM Almaden Research Center, 650 Harry Road, San Jose, California 95120-6099, and School of Mathematical Sciences, Tel Aviv University, Tel Aviv, Israel.

is considered the more natural representation. The complexity of solving the game depends, of course, on the representation, since the sizes of the two representations are not polynomially related. In this paper we discuss the extensive form. One node of the tree (the *root*) is distinguished as a starting point. Each nonterminal node is either a decision point for one of the players, or a *chance node* representing a random move. A path from the root to a terminal node is called a *play*. An edge leading from a node away from the root denotes a possible move for the corresponding player in case that node is reached during the play. Each chance node has a probability distribution over its moves. A payoff vector, containing each player's payoff, is associated with each terminal node. The two-player game is *zero-sum* if the sum of the payoffs to the two players at each terminal node is zero. The set of all decision nodes for a player is partitioned into *information sets*. When the player has to decide which move to take, he is aware of the information set he is at, but does not know the exact node within the set. To ensure that the player will not be able to deduce his exact location, the number of moves emanating from the node has to be the same for every node in the information set. The moves at different nodes of the same information set are therefore grouped into equivalence classes, which correspond to the possible decisions a player can make at that information set (each decision defines one move from each node in the information set). It is normally assumed that an information set is not visited more than once during one play.

A *pure strategy* for a player is an assignment of a decision to each of his information sets. A *behavior strategy* is an assignment of a probability distribution over the possible decisions at any information set, i.e., it is a plan for local randomization. A *mixed strategy* for a player is a probability distribution over all pure strategies for that player.

The game is said to have with *perfect information* if each information set consists of exactly one node. Zermelo's Theorem (see Zermelo [1913] and von Neumann and Morgenstern [1947]) asserts the existence of optimal pure strategies for two-person zero-sum games with perfect information. **?** (**?**) extended this theorem to n-person games with perfect information. The presence of nontrivial information sets complicates the game significantly. A "brute-force" method of solving a game in extensive form is to translate it into normal form. This entails listing all the possible pure strategies for each player, and the payoff for each strategy combination. The standard solution concept for two-player zero-sum games is the *saddle-point* (max-min strategies) which is a special case of the *Nash Equilibrium* for non-zero-sum games. The solution of a game in normal form is usually a mixed strategy for every player. In the case of two-person zero-sum games, optimal mixed strategies for the players can be found by solving a linear programming problem.

? (**?**) showed that for a large class of games, known as games with *perfect recall*, behavior strategies are "as good" as mixed strategies. Intuitively, a game with perfect recall is one in which each player does not forget anything and, in particular, remembers all his previous moves. In other words, at each point in the game, the player remembers

what information sets he has visited until that point. In such games, any mixed strategy induces a behavior strategy with the same payoffs. It can easily be shown that without perfect recall, payoffs which can be achieved with mixed strategies may not be achievable with behavior strategies (see Example 2.4). It seems that by playing a mixed strategy which cannot be described as a behavior strategy, a player might obtain more information than is allowed by his information partition. Thus, the feasibility of playing a mixed strategy contradicts the structure of the game. We are therefore interested in finding good behavior strategies for the case of imperfect recall.

Behavior strategies also have the advantage that less memory space is required for their implementation than mixed ones. The dimension of the mixed strategy vector is the number of available pure strategies, which may grow exponentially with the number of information sets. The dimension of the behavior strategy grows only linearly with this number. For example, Kuhn (1950) described a trivial poker game, in which the simplex of mixed strategies has dimension 8191 while the "cube" of behavior strategies has dimension 13. Clearly, finding optimal behavior strategies is more practical even in games with perfect recall. Since the mixed strategies induce behavior strategies, it is possible in principle to solve the normal form game to get an optimal mixed strategy, and compute the behavior strategy induced by it. Unfortunately, it may take exponential time to transform the extensive game into the normal form. It is therefore very desirable to have a technique for finding an optimal behavior strategy directly from the extensive form.

In this paper, we study the complexity of the problem of finding optimal (or max-min) behavior strategies for games with and without perfect recall. Throughout the paper, we restrict ourselves solely to two-person zero-sum games. In Section 2.2 we show that finding any type of max-min strategy, either pure, behavior, or mixed, for a player with imperfect recall, is NP-hard, even if the game has no chance moves and the other player has perfect information. If both players have imperfect recall, and chance moves are allowed, the problem of finding a max-min pure strategy is Σ_2^p-complete (see Garey and Johnson [1979]). In Section 2.3, we show that in a game with imperfect recall, the max-min behavior strategy may rely on irrational numbers. Thus, they cannot be computed exactly in any number of steps. In Section 3 we present an algorithm for finding optimal behavior strategies for games with perfect recall, which runs in time polynomial in the size of the game tree. This is the first method that can solve games without perfect information in time polynomial in the size of the game tree. The best techniques up to now have solved the game in time polynomial in the size of normal form, which is exponential in the size of the game tree.

2. Games without perfect recall

2.1 On mixed strategies and behavior strategies

Behavior strategies were first introduced by **?** (**?**). He also introduced the notion of imperfect recall, and showed that (i) in a game with perfect recall behavior strategies are "as good as" mixed strategies, and (ii) if not all players have perfect recall then there may exist mixed strategies which are "superior." Proposition 2.3 below is a simple generalization of the above.

We first introduce some notation. The set of pure strategies of player i is denoted by S_i. The set of information sets belonging to player i are denoted by $\mathcal{U}_i$. For $T \in \mathcal{U}_i$ and $s \in S_i$, let $s(T)$ denote the move prescribed by s at T. The set of behavior strategies of player i is denoted by B_i. For $\beta \in B_i$, we denote by $\beta_j(T)$ the conditional probability with which a move j is chosen under β, given that T is reached. If μ is a mixed strategy for player i, then for every $s \in S_i$, we denote by $\mu(s)$ the probability with which μ chooses strategy s. The expected payoff for player i under a combination of strategies $\boldsymbol{\mu} = (\mu_1, \mu_2)$ (where, in particular, each μ_k may be a pure or a behavior strategy) is denoted by $H_i(\boldsymbol{\mu})$.

The following definition is due to Kuhn:

Definition 2.1. Given a mixed strategy μ_i of player i, denote by Rel (μ_i) the family of all information sets $T \in \mathcal{U}_i$ such that there exists a mixed strategy μ_j for the other player, so that S is reached with positive probability when these strategies (μ_1 and μ_2) are played.

Definition 2.2. Given an arbitrary mixed strategy μ, we define the *induced behavior strategy* β as follows. For any move j, let

$$\beta_j(T) = \begin{cases} \left(\sum_{s:T\in Rel(s), s(T)=j} \mu(s)\right) \Big/ \left(\sum_{s:T\in Rel(s)} \mu(s)\right) & \text{if} \quad T \in \text{Rel } (\mu) \\ \sum_{s:s(T)=j} \mu(s) & \text{if} \quad T \notin \text{Rel } (\mu) \ . \end{cases}$$

Proposition 2.3. [**?** [**?**]] *A game tree has perfect recall if and only if for any payoff function H, for any pair of mixed strategies $\boldsymbol{\mu}$, and for each player i, if $\beta_i \in B_i$ is induced by μ_i, then*

$$H_1(\beta_1, \mu_2) = H_1(\mu_1, \mu_2)$$

and

$$H_2(\mu_1, \beta_2) = H_2(\mu_1, \mu_2) \ .$$

Figure 1: A game tree where player II has imperfect recall.

	$(1,1)$	$(1,2)$	$(2,1)$	$(2,2)$
(1)	-4	0	0	0
(2)	0	0	0	-4

Table 1: The normal form of the game.

Thus, in the case of perfect recall one might as well use behavior strategies, since they yield the same payoff and are computationally easier to handle.

In the case of imperfect recall, behavior strategies cannot always yield optimal payoffs, as can be seen in the following example:

Example 2.4. In the game tree of Figure 1, player II has imperfect recall. We denote the pure strategies of player I by $(s(T))$ and the pure strategies of player II by $(s(U_1), s(U_2))$. The normal form of the above game is shown in Table 1. The optimal strategy for player I is $\mu_1(1) = \mu_1(2) = \frac{1}{2}$. The optimal strategy for player II is:

$$\begin{aligned} \mu_2(1,1) = \mu_2(2,2) &= \tfrac{1}{2} \\ \mu_2(1,2) = \mu_2(2,1) &= 0 \end{aligned} \tag{1}$$

Therefore, using a mixed strategy, player II can assure an expected payoff of 2. On the other hand, if player II uses a behavior strategy β, with $\beta(U_1) = x$ and $\beta(U_2) = y$, then

$$H_2((1),(x,y)) \;=\; 4xy$$

$$H_2((2),(x,y)) \quad = \quad 4(1-x)(1-y)$$

The maximum expected payoff that player II can assure himself is:

$$\max_{x,y\in[0,1]} \min\{4xy, 4(1-x)(1-y)\} = 1 \ .$$

Note that if player II plays the mixed strategy shown in equation (1), then he recalls which pure strategy he is supposed to play and can therefore deduce the actual node he is at. Thus, using a mixed strategy gives player II more information than is permitted by the structure of the game. In order to remain within the information structure defined by the game, the player can use only behavior strategies. Therefore, from a worst-case optimization point of view, the best a player with imperfect recall can do is to use the max-min behavior strategy.

2.2 Finding max-min strategies

In this subsection we study the complexity of finding max-min strategies in a game where one of the players does not have perfect recall. In order to discuss the complexity of this problem, we must first define the length of the input.[1] Since the game is defined by the tree, including the information sets, the payoffs, and the probabilities associated with the random moves, we define the size of the game as follows. First, as usual, the size of an integer I is $\lceil \log_2(1+|I|) \rceil$. The size of a rational number is the sum of the sizes of its numerator and denominator in the reduced form. The size of a game equals the sum of the sizes of the payoffs and of the probabilities associated with random moves.

Proposition 2.5. *In a one-player game with chance moves, the problem of deciding whether the player can assure an expected payoff of at least λ (where λ is a given rational number) is in the class* NP.

Proof: There exists a pure strategy which maximizes the player's expected payoff relative to the entire set of mixed strategies. Thus it suffices to discuss pure strategies. It takes linear time to guess a pure strategy for the player and compute the expected payoff it yields. ∎

Proposition 2.6. *For any kind of strategies (either pure, behavior, or mixed), the problem of deciding whether player II can guarantee an expected payoff of at least λ, using the said kind of strategies, is* NP*-hard even if player I has perfect recall and there are no chance moves.*

[1] This definition will also serve us in Section 3 for analyzing the complexity of the algorithms presented there.

Proof: The proof goes by reduction from the 3-satisfiability problem (see Garey and Johnson [1979]). Given m clauses $x_i \vee y_i \vee z_i$ $(i = 1, \ldots, m)$, where $\{x_i, y_i, z_i\} \subset \{u_1, \bar{u}_1, \ldots, u_n, \bar{u}_n\}$, we construct a two-person zero-sum game as follows. Player I starts by choosing a number i where $1 \leq i \leq m$. Intuitively, player I chooses a clause. Let j_1, j_2 and j_3 denote indices such that

$$\{x_i, y_i, z_i\} \subset \{u_{j_1}, \bar{u}_{j_1}, u_{j_2}, \bar{u}_{j_2}, u_{j_3}, \bar{u}_{j_3}\} \ .$$

Player II is then informed of the members of $C = \{j_1, j_2, j_3\}$, one at a time, according to numerical order. Player II has to choose for each $j \in C$ either u_j or $\bar{u}_j$. Nonetheless, player II has only n information sets, corresponding to the indices $j = 1, \ldots, n$, so that when he is informed of an index j, he does not recall any of his previous choices or even the indices he has been presented with, if at all. Note that, intuitively, when player II is presented with j, he knows only that a clause which contains j has been chosen. Since he does not remember his previous moves, he does not have any other information about the clause. Since the player's decision at a node in information set j has to be the same for all clauses containing j, the player's decision at that information set corresponds exactly to an assignment of a truth-value to u_j. If the set of choices made by player II intersects the set $\{x_i, y_i, z_i\}$, then he receives a payoff of 1 from player I. Otherwise he receives 0. It is easy to see that player II can assure a payoff of 1 if and only if the conjunction of the given m clauses has a satisfying assignment. The size of the tree is $O(m)$.

To complete the proof for the cases of behavior and mixed strategies, note that since the maximum payoff in the game is 1, a probabilistic strategy which guarantees an expected payoff of 1 must be a probabilistic mixture of pure strategies each of which assures a payoff of 1. ∎

Proposition 2.7. *If player I has perfect recall or if there are no chance moves in the game, then the problem of deciding whether player II can guarantee a certain given rational payoff using pure strategies is* NP*-complete.*

Proof: In view of Proposition 2.6, we need only show that the problem is in NP. Now, we can guess a pure strategy s for player II in linear time. Consider the problem of computing a best response of player I to s. If player II plays s, then if player I has perfect recall, the game reduces to a one-player game with perfect recall. If there are no chance moves, the game becomes completely deterministic. Hence, for the problem of finding pure strategies, it can be treated as a game with perfect information. We will show later (see Section 3.3) that in these cases player I's problem can be solved in polynomial time, and hence the expected payoff guaranteed to player II by playing the guessed strategy s can be found in polynomial time. ∎

We have shown that finding max-min behavior strategies is NP-hard. We have also shown that finding an optimal mixed strategy is NP-hard. One should note that an optimal mixed strategy can be computed from the normal form of the game by solving a linear programming problem. This, however, takes more than polynomial time. Max-min behavior strategies can be found using nonlinear optimization techniques which will not be discussed in this paper.

Corollary 2.8. *It is* NP*-complete to decide, given a behavior strategy of player I and a rational number λ, whether there exists a pure strategy of player II which yields an expected payoff of at least λ.*

Proof: The problem is in NP since it takes linear time to guess a pure strategy for player II and calculate the expected payoff it yields. The NP-hardness follows from the construction in Proposition 2.6. ∎

Remark 2.9. This result is also true for mixed strategies if the input to the algorithm is a mixed strategy in sparse representation (i.e., the zero coordinates are not listed). It is shown in **?** (**?**) that every mixed strategy has an equivalent mixed strategy whose size is linear in the size of the game tree. These "small" mixed strategies can be used in the proof of Proposition 2.6.

In Proposition 2.7, we showed that in a certain class of games, the problem of determining whether player II can assure a certain payoff in pure strategies is NP-complete. The following Proposition shows that for games not in that class, the problem is complete for a higher complexity class (for the precise definition see Garey and Johnson [1979]).

Proposition 2.10. *The problem of deciding, in a game with chance moves where both players have imperfect recall, whether one of the players can assure a certain payoff λ, using* pure strategies*, is Σ_2^p-complete.*

Proof: The problem is in Σ_2^p, because we need only check whether there exists a pure strategy s_1 for player I such that for any pure strategy s_2 of player II, the expected payoff under s_1 and s_2 is greater than λ. Since both s_1 and s_2 have size polynomial in the size of the tree, this can be done in Σ_2^p.

A canonical Σ_2^p-complete problem is the following: given a disjunction of k clauses $(x_i \wedge y_i \wedge z_i)$, where

$$\{x_i, y_i, z_i\} \subseteq \{u_1, \bar{u}_1, \ldots, u_n, \bar{u}_n, v_1, \bar{v}_1, \ldots, v_m, \bar{v}_m\}$$

determine whether

$$(\exists u_1, \ldots, u_n)(\forall v_1, \ldots, v_m) \bigvee_{i=1}^{k} (x_i \wedge y_i \wedge z_i) \ . \tag{2}$$

The proof is by reduction to this problem. We construct a two-person zero-sum game as follows. Initially, the chance player randomly chooses i, where $1 \leq i \leq k$ is a clause. Let C and D denote the sets of indices of existentially and universally (respectively) quantified variables which appear in clause i. Player I is then informed of the indices $j \in C$ one by one, and has to choose a value of value of either 0 or 1 for each of them. As in Proposition 2.6, the player has only n information sets, one for each $j = 1, \ldots, n$. Thus, player I cannot determine which clause was picked by chance. After player I makes his choice, player II goes through the same procedure for the indices $\ell \in D$. Since player II has m information sets, one for each $\ell = 1, \ldots, m$, he cannot determine the clause picked by chance, nor the choices made by player I. If the values for the variables in clause i, as chosen by the two players, give a value of 1 to the clause chosen by chance, then player I gets a payoff of 1. Otherwise, both players get a payoff of 0.

In the game described above, player I can assure a payoff greater than 0 if and only if equation (2) is true. Assume that equation (2) is true, and let $\boldsymbol{u} \in \{0,1\}^n$ be a satisfying assignment of truth-values to $u_1, \ldots, u_n$. Suppose player I uses $\boldsymbol{u}$ as his strategy. No matter what player II's strategy $\boldsymbol{v}$ is, there will always be some clause i which is true under $\boldsymbol{u}$ and $\boldsymbol{v}$. This clause will be chosen by chance with positive probability, and therefore, there is a positive probability that player I will get a positive payoff. This implies that player I's expected payoff is positive. Conversely, suppose that (2) is false, and let $\boldsymbol{u}$ be a strategy for player I. There exists a truth-value assignment $\boldsymbol{v} \in \{0,1\}^m$ for $v_1, \ldots, v_m$, such that all clauses are false under $\boldsymbol{u}$ and $\boldsymbol{v}$. If player I plays $\boldsymbol{u}$ then player II may play $\boldsymbol{v}$. In this case all the clauses are false, and therefore all of chance's moves give player I a payoff of 0. Thus, player I cannot assure a positive payoff. ∎

Remark 2.11. Unlike Proposition 2.6, the proof of Proposition 2.10 extends neither to mixed nor to behavior strategies, since it relies on the assumption that player II's strategy is chosen based on knowledge of player I's strategy.

2.3 A max-min behavior strategy may require irrational numbers

In a two-person zero-sum game with perfect recall, if the payoffs are rational numbers, then there exist optimal mixed strategies which use only rational probabilities. Thus, the players also have max-min behavior strategies using rational numbers. It is interesting to observe that this is not true in games with imperfect recall, as shown in the following example:

Figure 2: The game tree of a game with imperfect recall for which the optimal behavior strategy requires irrational numbers.

Example 2.12. Consider a two-person zero-sum game where player I chooses $1, 2$ or 3. Player II is not informed of player I's choice when he has to choose first between l or r and then, without recalling his first choice, between L and R. The payoff to player II is 3 if the choices are either $(1,(l,L))$ or $(2,(r,R))$, 1 if they are either $(3,(l,R))$ or $(3,(r,L))$, and 0 otherwise. The game tree is portrayed in Figure 2.

Denote by x and y the probabilities of choosing l and L, respectively, and by $\bar{x} = (1-x)$ and $\bar{y} = (1-y)$ the probabilities of choosing r and R, respectively. Thus, player II's problem is to find $x, y \in [0,1]$ so as to maximize the following function:

$$P(x,y) = \min \{\ 3xy\ ,\ 3\bar{x}\bar{y}\ ,\ x\bar{y} + \bar{x}y\ \}\ .$$

It can be verified that

$$P(x,y) = \begin{cases} 3xy & \text{if} \quad x + y \leq 1 \text{ and if } \ x > \frac{1}{5} \text{ then } \ y \leq x/(5x-1) \\ 3\bar{x}\bar{y} & \text{if} \quad x + y \geq 1 \text{ and if } \ x < \frac{4}{5} \text{ then } \ y \geq (3-4x)/(4-5x) \\ x\bar{y} + \bar{x}y & \text{if} \quad \frac{1}{5} < x < \frac{4}{5} \text{ and } \ x/(5x-1) \leq y \leq (3-4x)/(4-5x) \end{cases}$$

The function $P(x,y)$ is shown in Figure 3. It can easily be seen that the max-min is attained when

$$3xy = 3(1-x)(1-y) = x(1-y) + (1-x)y\ .$$

This implies $3x(1-x) = x^2 + (1-x)^2$, so $5x^2 - 5x + 1 = 0$ and $x = 0.1(5 \pm \sqrt{5})$.

Figure 3: A three dimensional image of $P(x, y)$, the payoff guaranteed to player II when using the strategy pair (x,y).

The fact that the max-min behavior strategies require irrational numbers has implications with regard to the complexity of finding them. The max-min behavior strategy cannot be computed by any algorithm in one of the standard complexity classes. It requires the ability to solve algebraic equations. Nonetheless, the complexity of deciding whether a player can assure a certain payoff λ (which was discussed in the previous section) is meaningful in the usual model of computation.

3. Players with perfect recall

In this section, we discuss the complexity of computing the max-min strategy in a two-person zero-sum game with perfect recall which is given in the extensive form. By Kuhn's theorem (see Section 2.1), in a game with perfect recall, behavior strategies are as good as mixed strategies. Therefore, the game has a *saddle point* in behavior strategies.

It is well known that a saddle point can be computed as follows. First, enumerate all the pure strategies of both players, and generate the payoff matrix $\boldsymbol{A} = (a_{ij}) \in R^{m \times n}$, where m and n are the numbers of pure strategies of player I and player II, respectively. More precisely, let a_{ij} be the amount that player I pays to player II when they choose their i'th and j'th pure strategies, respectively. Player II is then looking for a vector $\boldsymbol{x} \in R^n$ such that $\sum_{i=1}^{n} x_i = 1$ and $\boldsymbol{x} \geq \boldsymbol{0}$, so as to maximize λ, subject to $\boldsymbol{Ax} \geq \lambda \boldsymbol{e}$ (where $\boldsymbol{e} = (1, \ldots, 1)^T \in R^m$). Thus, player II solves a linear programming problem with $n + 1$ variables and m inequalities. Since both m and n may grow exponentially in the size of the game tree, this solution may require exponential time. On the other hand, since the linear programming problem can be solved in polynomial time, it follows that this method does not require more than exponential time.

If necessary, the behavior-strategy equilibrium can then be obtained by converting the mixed strategies into behavior strategies. The number of variables describing a behavior strategy is polynomial in the size of the game tree. Thus, it is conceivable that a saddle-point in behavior strategies can be computed directly, in polynomial time, without the construction of mixed strategies. Our goal here is to show that if the game has perfect recall (but not necessarily perfect information), then the problem can be solved in polynomial time.

The max-min behavior strategy cannot be directly translated into a linear programming problem, since the payoff is not a linear function of the probabilities that describe the behavior strategy. Thus, it is not clear whether the problem can be formulated as a linear programming problem with dimensions that are bounded by a polynomial in the size of the game tree. Nonetheless, we will show in the next section that it can be formulated as a linear programming problem with a polynomial number of variables and an exponential number of constraints. Problems of this sort can sometimes be solved using the ellipsoid method (as in **?** [?]) in polynomial time despite the fact that the

number of constraints is exponential. This can be done when the constraints are not listed explicitly, but rather generated dynamically as the need arises. The algorithm is described in Sections 3.3 and 3.4.

3.1 Perfect recall and structured information sets

If a player has perfect recall then his information sets must have a certain structure. In this section, we show that the information sets of a player with perfect recall form a *forest*, i.e., a set of disjoint trees. In Section 3.5, we continue to investigate the relationship between the structure of the information sets and the complexity of the resulting game.

If U and V are two information sets of a player i, we say that U *precedes* V, and denote $U \prec V$, if there exist vertices $u \in U$ and $v \in V$ such that u is the last decision node of player i on the path from the root to v. An information set which is not preceded by any other information set is called *initial*, and one which does not precede any other information set is called *terminal*.

Proposition 3.1. *The relation $\prec$ induces a forest on $\mathcal{U}_i$.*

Proof: It suffices to show that each information set has at most one "parent," i.e., there do not exist distinct information sets U, U', and V of player i, such that $U \prec V$ and $U' \prec V$. Suppose, to the contrary, that such information sets exist. Let $u \in U$ and $v \in V$ be nodes such that u is the last decision node of player i on the path from the root to v, and let $u' \in U'$ and $v' \in V$ be nodes such that u' is the last decision node on the path from the root to v'. Since there is a unique path from the root to v, and only one of u and u' ($u \neq u'$) can be the last node of player i on this path, it follows that $v \neq v'$. Thus, there are two distinct paths from the root into V, passing through two distinct information sets of player i. This contradicts the assumption of perfect recall. ∎

Corollary 3.2. *For a player with perfect recall, the relation $\prec$ induces a partial order on the information sets.*

Corollary 3.3. *For a player with perfect recall, there exist at least one initial information set, and at least one terminal information set.*

3.2 Formulating the set of behavior strategies

We first need to show how to formulate the optimization problem of finding a max-min behavior strategy as a *linear programming problem*. We define a vector $\boldsymbol{x} = (x_1, \ldots, x_p)^T$

of a polynomial number of *nonnegative* decision variables which describe (in a manner to be explained below) a behavior strategy for player II. These variables have the property that for any pure strategy s of player I, there exists a linear function $f_s(\boldsymbol{x}) = \boldsymbol{a}_s^T \boldsymbol{x}$ whose value is the payoff to player II when player I uses strategy s and player II uses the behavior strategy described by $\boldsymbol{x}$.

We now develop the system of constraints which $\boldsymbol{x}$ has to satisfy. Recall that the moves emanating from a node v are indexed by numbers $1, 2, \cdots$. Let $k(v)$ denote the number of possible moves at v. Suppose $U = \{u_1, \ldots, u_r\}$ is an information set. Necessarily, $k(u_1) = \cdots = k(u_r)$, and we can denote $k(U) = k(u_1)$. An index d $(1 \leq d \leq k(u))$ of a move emanating from a decision node u of player i is called a *decision* of that player. Note that in any pure strategy of player i, the decision for all the nodes in the same information set must be the same. We say that a node u is *ruled out* by strategy s of player I, if on the path from the root to u, there exists at least one decision of player I which is not taken under strategy s.

For each node u, let $\alpha(u)$ denote the product of all the probabilities corresponding to random moves along the path from the root to u, if any; otherwise, let $\alpha(u) = 1$. Similarly, for a behavior strategy β, let $\beta(u)$ denote the product of all the probabilities assigned by β to the decisions of player II appearing on the path from the root to u.

Given any pure strategy s of player I and any behavior strategy β of player II, for any node u, there is a well-defined probability $\pi(u; \beta, s)$ that the node u will be reached during the play of the strategies s and β.

Proposition 3.4. *If a node u is not ruled out by strategy s of player I, then*

$$\pi(u; \beta, s) = \alpha(u)\ \beta(u)\ .$$

This probability is therefore independent of s.

Proof: Suppose u is not ruled out by s. In this case $\pi(u; \beta, s)$ is equal to the product of the probabilities assigned by β to the decisions of player II which lie on the path from the root to u, and the probabilities corresponding to chance (i.e. nature's) moves on the tree along the same path. ∎

In view of Proposition 3.4, let $\pi(u; \beta)$ denote the probability that node u will be reached, provided it is not ruled out by player I.

Fact 3.5. *In a game with perfect recall, if u_1 and u_2 are nodes in the same information set of player II, then*

$$\beta(u_1) = \beta(u_2)\ .$$

Moreover, if d is a decision at u_1 and u_2, leading to v_1 and v_2 respectively, then

$$\beta(v_1) = \beta(v_2)\ .$$

Proof: The assumption of perfect recall implies that the sets of probability values assigned by β along the path leading from the root to u_1 and u_2 are identical. And since u_1 and u_2 are in the same information set, they must assign the same probability to the decision d. ∎

If u is a decision node of player II and d is a decision at u, let (u, d) denote v, where v is the child of u reached by taking the decision d.

We assign a variable x_u to each node u in the game tree. Intuitively, x_u denotes $\beta(u)$, where β is the corresponding behavior strategy for player II.

The constraints on the variables x_u are derived as follows:

- Let u_0 be the initial node. Then

$$x_{u_0} = 1 . \tag{3}$$

- Let $U = \{u_1, \ldots, u_r\}$ be any information set of player II. The information set U imposes constraints as follows. For every d $(d = 1, \ldots, k(U))$,

$$x_{(u_1,d)} = x_{(u_2,d)} = \cdots = x_{(u_r,d)} . \tag{4}$$

  This constraint enforces the second half of Fact 3.5. Intuitively, it means that the probability of taking any decision d must be the same at all nodes in the information set.
- Let U be an information set of player II, and let u be an arbitrary node in U. Then we also require that:

$$\sum_{d=1}^{k(U)} x_{(u,d)} = x_u . \tag{5}$$

  This constraint forces consistency between the probability of reaching a node and the probabilities of reaching its children. Note that because of equation (4), equation (5) is actually true for every $u \in U$.
- For any node v not belonging to player II, the decision made at v must not affect the variables. Recall that x_u is defined to depend only on player II's decisions, and not on chance decisions or decisions belonging to player I. We therefore require

$$x_v = x_{(v,1)} = x_{(v,2)} = \cdots = x_{(v,k(v))} . \tag{6}$$

We note that not all of these equations will be independent, but a simple inductive argument shows that the number of equations, plus the number of variables needed to describe the behavior strategy, equals the dimension of the vector $\boldsymbol{x}$.

Proposition 3.6. *There is a one-to-one correspondence between nonnegative vectors $\boldsymbol{x}$ satisfying equations* (3), (4), (5), *and* (6) *on the one hand, and behavior strategies β of player II on the other hand.*

Proof: We describe the correspondence as follows. Given a behavior strategy β, let $x_u = \beta(u)$. Equation (4) then follows from Fact 3.5. The rest of the equations follow immediately from the definitions.

Given an $\boldsymbol{x}$ which satisfies all the constraints, define the behavior strategy β associated with $\boldsymbol{x}$ as follows. Let U be an information set of player II, and let d ($d = 1, \ldots, k(U)$) be any decision. Choose an arbitrary $u \in U$, and define

$$\beta_d(U) = \frac{x_{(u,d)}}{x_u} \ .$$

This is well-defined and independent of u due to equation (4). Due to equation (5), we obtain that

$$\sum_{d=1}^{k(U)} \beta_d(U) \ = \ \frac{\sum_{d=1}^{k(U)} x_{(u,d)}}{x_u} \ = \ 1 \ .$$

It is easy to verify that for this β, $\beta(z) = x_z$ for all z. Equation (3) guarantees that only a single vector $\boldsymbol{x}$ corresponds to each behavior strategy. We have thus defined a one-to-one correspondence. ∎

The construction of the vector $\boldsymbol{x}$ and the constraints is illustrated in the following example.

Example 3.7. Consider the game tree in Figure 4, where both players have perfect recall. We will represent the set of behavior strategies of player II with linear inequalities. Consider a vector $\boldsymbol{x} = (x_1, \ldots, x_{36})^T$ whose components correspond to the nodes of the tree. Equation (3) implies that:

$$x_1 \ = \ 1 \ . \tag{7}$$

The information set U_1 implies the following constraints:

$$x_5 \ = \ x_6 \tag{8}$$

$$x_{12} \ = \ x_7 \ . \tag{9}$$

and that

$$x_5 + x_6 \ = \ x_2 \ . \tag{10}$$

The set U_2 is a singleton set, so it implies only the constraint:

$$x_8 + x_9 \ = \ x_4 \ . \tag{11}$$

The information set U_3 implies the following constraints:

$$x_{10} \ = \ x_{24} \tag{12}$$

$$x_{11} \ = \ x_{25} \tag{13}$$

$$x_{10} + x_{11} \ = \ x_5 \ . \tag{14}$$

Figure 4: A game tree with perfect recall.

The information set U_4 implies the constraints:

$$x_{26} = x_{28} \tag{15}$$
$$x_{27} = x_{29} \tag{16}$$
$$x_{26} + x_{27} = x_{13} \ . \tag{17}$$

The set U_5 also implies only the constraint:

$$x_{31} + x_{32} = x_{15} \ . \tag{18}$$

For U_6,

$$x_{33} = x_{35} \tag{19}$$
$$x_{34} = x_{36} \tag{20}$$
$$x_{33} + x_{34} = x_9 \ . \tag{21}$$

Each of the nodes of player I and the chance node (nodes 1,8,9,10,11,12, and 7) also implies constraints:

$$x_1 = x_2 = x_3 = x_4 \tag{22}$$

$$x_8 = x_{30} = x_{15} \tag{23}$$
$$x_9 = x_{16} = x_{17} \tag{24}$$
$$x_{10} = x_{18} = x_{19} \tag{25}$$
$$x_{11} = x_{20} = x_{21} \tag{26}$$
$$x_{12} = x_{22} = x_{23} \tag{27}$$
$$x_7 = x_{13} = x_{14} \ . \tag{28}$$

Remark 3.8. Note that most of the equations defined by this process are equations forcing the equality of two different variables, since the paths to the two nodes pass through exactly the same information sets of player II. Since the complexity of the algorithm presented in Section 3.4 depends heavily on the number of variables, we may be able to significantly reduce its running time by unifying variables guaranteed to be equal. This unification process can be carried out in polynomial time.

The optimization problem of player II calls for an $\boldsymbol{x}$ which maximizes the minimum of $f_s(\boldsymbol{x})$ over all pure strategies s of player I. In our construction, the functions $f_s(\boldsymbol{x})$ take the form:

$$f_s(\boldsymbol{x}) = \sum_{z \in F(s)} \alpha(z)\ a_z\ x_z \ ,$$

where a_z is the payoff to player II at a terminal node z, and $F(s)$ denotes the set of terminal nodes which are not ruled out by s. The functions corresponding to the game of Figure 4 are listed below.

Example 3.9. In the game of Example 3.7, player I has five pure strategies:

$$\begin{aligned}
s_1 &: \ d(T_1) = 1 \ , \ d(T_2) = 1 \\
s_2 &: \ d(T_1) = 1 \ , \ d(T_2) = 2 \\
s_3 &: \ d(T_1) = 2 \\
s_4 &: \ d(T_1) = 3 \ , \ d(T_3) = 1 \\
s_5 &: \ d(T_1) = 3 \ , \ d(T_3) = 2 \ .
\end{aligned}$$

Thus, the linear programming problem is:

$$\begin{aligned}
\text{Maximize} \quad & \lambda \\
\text{subject to} \quad & \lambda \le a_{18}x_{18} + a_{20}x_{20} + a_{22}x_{22} \\
& \lambda \le a_{19}x_{19} + a_{21}x_{21} + a_{23}x_{23} \\
& \lambda \le a_{24}x_{24} + a_{25}x_{25} + p\ a_{26}x_{26} + p\ a_{27}x_{27} + (1-p)\ a_{28}x_{28} + (1-p)\ a_{29}x_{29} \\
& \lambda \le a_{30}x_{30} + a_{33}x_{33} + a_{34}x_{34} \\
& \lambda \le a_{31}x_{31} + a_{32}x_{32} + a_{35}x_{35} + a_{36}x_{36} \\
& x \in X \ ,
\end{aligned}$$

where X denotes the set of nonnegative vectors $\boldsymbol{x}$ which satisfy (7)–(28).

The number of pure strategies of player I may grow exponentially with the size of the tree. Given a vector $\boldsymbol{x}$, however, there is a polynomial-time procedure for finding a pure strategy s which minimizes $f_s(\boldsymbol{x})$. The procedure is polynomial time regardless of the number of strategies of player I, since it avoids listing them. Rather, it directly computes the minimizing strategy s, which is actually player I's "best response" to $\boldsymbol{x}$. The algorithm for finding a best response to a given behavior strategy is given in the following section.

3.3 Best response by a player with perfect recall

Given any game tree where player I has perfect recall, we can find the best response of player I to any mixed strategy of player II. **?** (**?**) shows that when player II's mixed strategy is fixed, the game reduces to a one-player game (only player I playing) with chance moves, where all the information sets are *singletons*. We will give a simple method for computing a best response from the original game tree and the mixed strategy of player II.

Using the mixed strategy of player II, it is easy to compute for every node u the probability $\pi(u)$ of reaching that node, assuming player I plays a pure strategy which does not rule u out. Our calculation of a best response pure strategy proceeds bottom up, from the terminal nodes to the root, using the partial order induced on the information sets of player I (as in Proposition 3.1). Concurrently with the computation of the best response strategy, we also compute for each node u, a value $h(u)$. This represents the conditional expected payoff to player I, given that u is reached and that a best response strategy is used in the subgame originating at u.

The construction proceeds as follows:

- For every terminal node z,
$$h(z) = H_1(z) \ .$$
- Let u is a node not belonging to player I, such that h and s have been computed for all of u's children. The node u is either a chance node or a node of player II. In either case, we can compute $p_d(u) = \pi(u,d)/\pi(u)$, which is the probability that the decision d is taken at u given that u is reached. We define
$$h(u) = \sum_{d=1}^{k(u)} p_d(u)\ h(u,d) \ .$$
- Let $V = \{v_1, \ldots, v_r\}$ be an information set belonging to player I, such that h and s have been computed for all the children of any node in V. Since player I has perfect

recall, his own previous decisions cannot influence which node in V is reached, but only whether V is reached at all. In other words, a node $v \in V$ is ruled out if and only if every node $v' \in V$ is ruled out. Therefore, no matter what strategy player I uses, the unconditional probability of reaching the nodes in V will either be $\pi(v)$ for all $v \in V$ or 0 for all $v \in V$. In the second case, the decision at V has no effect. Thus, the decision at V can be made using the probabilities $\pi(v)$, independently of any decision that will be made later on. We therefore choose $s(V)$ to be the decision d, $1 \leq d \leq k(V)$, which maximizes the expression

$$\sum_{i=1}^{r} \pi(v_i)\, h(v_i, d) \ .$$

We then define

$$h(v_i) = h(v_i, d) \ .$$

The complexity of dealing with each node is linear in the number of edges emanating from the node under consideration. Therefore, the algorithm runs in time linear in the size of the tree.

3.4 Computing the max-min behavior strategy

In Section 3.2 we showed how to formulate the max-min behavior strategy of a player with perfect recall as a linear programming problem. The number of constraints in our representation of the problem is equal to the number of pure strategies of the other player. This can be exponential in the size of the game tree, which would seem to entail an exponential algorithm.

Fortunately, our problem belongs to a class of combinatorial optimization problems which have polynomial-time "separation oracles." Although we have exponentially many inequalities, there exists a polynomial time algorithm $\mathcal{A}$ that tests whether a given point satisfies all the inequalities, and if not, finds a violated one. For this class of problems, the ellipsoid method can be applied to solve the combinatorial optimization problem in polynomial time (see **?** [**?**]) as we show below.

In our problem, there are two types of inequalities. First, we have the equations that constrain the vector $\boldsymbol{x}$ to describe a behavior strategy of player II. We also have those inequalities which model the requirement that this strategy guarantee a certain value against all strategies of player I, as described above. The separation oracle needs to check if all the inequalities are satisfied, and if not, to find a violated inequality. For the inequalities of the first type, this test is easily accomplished in polynomial time. For the inequalities of the second type, this is essentially the best response algorithm described in the previous section. It finds a best response s of player I to a certain behavior strategy,

modeled by $\boldsymbol{x}$, of player II. If the payoff to player II is at least the current λ, then all the inequalities of the second type are satisfied. If not, then the inequality associated with s is a violated inequality. Based on the separation oracle defined above, the ellipsoid algorithm gives us a polynomial time algorithm, in the size of the game tree, for finding max-min strategies in a game where both players have perfect recall.

3.5 Perfect memory and partially ordered information sets

We showed in Proposition 3.1 that the relation $\prec$ gives a forest structure to the family of information sets of a player with perfect recall. On the other hand, this relation may be form a forest even when the player does not have perfect recall. For example, in the game of Example 2.12, player II has two information sets, one of which is the child of the other, but does not have perfect recall.

We can provide some intuition for the general case of a forest of information sets. We say that a player whose family of information sets forms a forest has *perfect memory*. Games with perfect memory were first defined by Okada (1987). Perfect memory means that even though a player may not remember his previous moves, he never forgets anything he once knew. A number of multi-agent games have such a structure.

It is interesting to investigate the complexity of finding max-min behavior strategies for games with perfect memory. Example 2.12 shows that even for a player with perfect memory, a max-min behavior strategy may use irrational numbers. Thus, it cannot be formulated as a linear programming problem. On the other hand, the NP-hardness proof no longer holds for this case, as the information sets used in Proposition 2.6 require a more complex information structure. In particular, Corollary 2.8 no longer applies.

Games with perfect memory form a subclass of the class of games whose complete inflation has perfect recall.

The term "complete inflation" was defined by Dalkey (1953). An *immediate inflation* of an information partition is a partition where one of the information sets has been split, so as to enable the player to remember one of his own decisions. An information partition $\mathcal{U}$ is an *inflation* of $\mathcal{U}'$ if there exist $\mathcal{U}_1, \ldots, \mathcal{U}_k$ such that $\mathcal{U}_1 = \mathcal{U}$ and $\mathcal{U}_k = \mathcal{U}'$ and $\mathcal{U}_{i+1}$ is an immediate inflation of $\mathcal{U}_i$ for $i = 1, \ldots, k-1$. An information partition $\mathcal{U}'$ is a *complete inflation* of $\mathcal{U}$ if it is an inflation of $\mathcal{U}$ and does not have an immediate inflation.

Dalkey (1953) shows that inflating the information partitions in a game produces an equivalent game, i.e., for any pure strategy combination in the inflated game, there exists a pure strategy combination in the original game which gives the same payoff. Based on this result, our best response algorithm of Section 3.3 can be modified to handle games whose complete inflation has perfect recall, and in particular, games with perfect memory. Recall that the algorithm initially takes a game tree and a mixed strategy for

one of the players, and creates a one-person game tree for the other player, with the same information structure. If the player has perfect memory, then the following technique can be used. The game is gradually inflated until it is completely inflated. According to Okada's paper, the complete inflation of games with perfect memory has perfect recall. A best-response pure strategy is then found for this new game. According to Dalkey's result, there is a pure strategy in the original game which yields the same payoff. Since the set of strategies in the original game is a subset of the set of strategies in the inflated game, this strategy must be optimal in the original game.

Below we give algorithms for finding the complete inflation of a game and for transforming strategies in the inflated game to equivalent strategies, i.e., strategies yielding the same payoffs, in the original game.

Algorithm 3.10. Traverse each tree in the forest of information sets from the root down. The root information set is left as is. The algorithm proceeds recursively. Assume that all the information sets on the path from the root to the current information set U have been partitioned. The set U may have several parent information sets, say $V_1, \ldots, V_k$. We therefore partition U into subsets $U_1, \ldots, U_k$, so that each subset includes children of only one information set. Specifically, let U_i be the set of nodes $v \in U$ which are descendants of a node in V_i. This preserves the tree structure. For each new information set U_i, and for each decision d, $(d = 1, \ldots, k(V_i))$, we define U_i^d to be the set of nodes $v \in U_i$ which are reached from V_i via decision d. These subsets form the final partition of U. The process continues recursively. It is clear that this algorithm is polynomial in the number of nodes in the game tree (but not in the number of original information sets).

Lemma 3.11. *The information partition $\mathcal{U}'$ defined in Algorithm 3.10 is the complete inflation of the original partition $\mathcal{U}$.*

Proof: Each step in the algorithm inflates the original partition. Since the final result has perfect recall, it follows from the assumption of perfect memory that this has to be the complete inflation. ∎

We now run the algorithm for computing a best response on $\mathcal{U}'$, to get a pure strategy s' which is a best response to the mixed strategy μ of player II. As explained above, s' induces a pure strategy s on the original game, with identical payoff. The construction of s proceeds from the roots in the forest of information sets of player I down to the leaves. Let U be a root in the forest. Since U was not partitioned, we can define $s(U) = s'(U)$. The construction continues inductively. Let U be an information set such that for all ancestors V of U, $s(V)$ has been defined. Let T be the set of nodes in U which are not ruled out by the partial strategy s. Note that the already determined parts of s are sufficient to determine which nodes in U the strategy s rules out. It is clear from the construction of $\mathcal{U}'$ that $T \in \mathcal{U}'$. We define $s(U) = s'(T)$. The fact that the original

information partition is a forest is crucial here. Otherwise, there might have been two paths from the root information set to U, causing T to contain more than one of the information sets in $\mathcal{U}'$. In that case, a conflict would have arisen in the definition of $s(U)$. The process now continues recursively.

The payoff associated with s is identical to the payoff associated with s', because both strategies rule out exactly the same set of nodes in the tree.

The construction presented above, together with the best response algorithm from Section 3.3, extends the result of the previous section to cases where not both players have perfect recall. If player II has perfect recall and player I has perfect memory, there is still a polynomial time algorithm for finding max-min behavior strategies for player II.

4. Conclusion

We have examined the complexity of finding max-min strategies in two-person zero-sum games in extensive form. Until now, the work on two-person zero-sum games has concentrated mostly on the normal form and on mixed strategies. This approach entails the transformation of every game into the normal form, which is frequently exponential in the size of the original game tree. To avoid this exponential process, it is necessary to obtain a solution directly from the game tree. Since the mixed strategy solution also requires exponential size, a different solution concept, the behavior strategy, must be used.

The results of this paper map the complexity of this problem in a variety of cases. The complexity is shown to be directly related to the memory of the two players have. When both players have perfect recall, the solution of the game is computable in polynomial time. If at most one player has perfect recall, the problem is NP-complete. If none of the players has perfect recall, and if chance moves are allowed, the complexity is higher. Table 2 shows the complexity of the problem under different assumptions, as known to date.

References

[Dalkey, 1953] N. Dalkey. Equivalence of information patterns and essentially indeterminate games. In H. W. Kuhn and A. W. Tucker, editors, *Contributions to the Theory of Games II*, pages 217–243. Princeton University Press, Princeton, 1953.

[2]The best algorithm known to date takes exponential time. The precise complexity class of this problem is unknown.

Player I	Player II	Chance	Strategy	Complexity	Reference
perfect information	perfect information	yes	pure	polynomial	Zermelo's Theorem
perfect information	imperfect recall	no	all types pure	NP-hard NP-complete	Proposition 2.6 Proposition 2.7
imperfect recall	imperfect recall	yes	pure	Σ_2^p-complete	Proposition 2.10
imperfect recall	imperfect recall	yes	mixed	exp. time[2]	standard alg.
perfect memory	perfect recall	yes	behavior	polynomial	Section 3

Table 2: The complexity of finding the max-min strategy for Player II, under various conditions. With regard to the hardness results, the table gives the complexity of deciding whether Player II can assure a certain payoff λ.

[Deng and Papadimitriou, 1989] X. Deng and C. H. Papadimitriou. The complexity of cooperative solution concepts. Unpublished Manuscript, 1989.

[Garey and Johnson, 1979] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness.* W. H. Freeman, San Francisco, 1979.

[Gilboa and Zemel, 1989] I. Gilboa and E. Zemel. Nash and correlated equilibria: Some complexity considerations. *Games and Economic Behavior*, 1:80–93, 1989.

[Kuhn, 1950] H. W. Kuhn. Extensive games. *Proc. National Academy of Sciences of the U.S.A.*, 36:570–576, 1950.

[Lucas, 1972] W. F. Lucas. An overview of the mathematical theory of games. *Management Science*, 15, Appendix P:3–19, 1972.

[McKinsey, 1952] J. McKinsey. *Introduction to the Theory of Games.* The RAND Series. McGraw-Hill, New York, 1952.

[Okada, 1987] A. Okada. Complete inflation and perfect recall in extensive games. *International Journal of Game Theory*, 16:85–91, 1987.

[Owen, 1982] G. Owen. *Game Theory.* Academic Press, New York, 1982.

[von Neumann and Morgenstern, 1947] J. von Neumann and O. Morgenstern. *The Theory of Games and Economic Behavior.* Princeton University Press, Princeton, 2nd edition, 1947.

[Zermelo, 1913] E. Zermelo. Über eine Anwendung der Mengenlehre auf die Theorie des Schachspiels. In E. W. Hobson and A. E. H. Love, editors, *Proc. 5th International Congress of Mathematicians II*, pages 501–504. Cambridge University Press, Cambridge, 1913.