Fast algorithms for finding proper strategies in game trees

Peter Bro Miltersen* and Troels Bjerre Sørensen*

Abstract

We show how to find a *normal form proper* equilibrium in behavior strategies of a given two-player zero-sum extensive form game with imperfect information but perfect recall. Our algorithm solves a finite sequence of linear programs and runs in polynomial time. For the case of a perfect information game, we show how to find a normal form proper equilibrium in linear time by a simple backwards induction procedure.

1 Introduction

It is well known that Nash equilibria of *matrix games* (i.e., two-player zero-sum games in normal form) coincide with pairs of minimax and maximin mixed strategies and can be found efficiently using linear programming. However, in many realistic situations where it is desired to compute prescriptive strategies for games with hidden information, the game is given in *extensive form*, i.e., as a *game tree* with a partition of the nodes into *information sets*. Each information set describes a set of nodes mutually indistinguishable for the player to move. One may analyze an extensive form game by converting it into normal form and then analyzing the resulting matrix game. However, the conversion from extensive to normal form incurs an exponential blowup in the size of the representation. Koller, Megiddo and von Stengel [10] showed how to use the *sequence form* representation to compactly represent and efficiently compute maximin behavior strategies for two-player extensive-form zero-sum games with imperfect information but perfect recall by solving linear programs of size linear in the size of the game trees, and avoiding the conversion to normal form. The method of Koller, Megiddo and von Stengel has been used for constructing prescriptive strategies for concrete, often very large games, to be used in game playing software. In particular, it was applied to solve various variants of two-player poker [1, 8] containing millions of information sets. The efficient algorithm, based on the sequence form representation and implemented using state of the art linear programming software was essential for obtaining solutions for games this large.

Despite its widespread use, the strategies computed by the algorithm suffer from a certain deficiency, first pointed out by Koller and Pfeffer [11]: While the strategy computed by the algorithm is a correct maximin strategy and thus guaranteed to attain an expected payoff of at least the game-theoretic value of the game considered against any counter strategy, it does *not* necessarily prescribe sensible play in any particular situation encountered during the game. Indeed, since the strategy computed is not attempting to achieve a payoff better than the value of the game, a player playing by the strategy will gladly give back any "gift" he receives from his opponent. The deficiency may be illustrated even with perfect information games. Consider the two perfect information games in Figure 1 (payoffs at leaves are paid by Player 2 to Player 1). The value of the game (a) is 0 and the strategy profile (i.e., a strategy for each player) choosing the actions indicated form a Nash equilibrium. However, if Player 1 gets to move, he should surely not perform the action U indicated, as the mistake of Player 2 has enabled him to achieve a payoff of 1 by choosing action D. Similarly, the value of the game (b) is 0 and the choices indicated form a Nash equilibrium. However, it does not seem sensible for Player 1 to open the game by choosing U as indicated. Rather than immediately achieving the payoff of 0, he should choose D, let Player 2 move and hope that Player 2 will make a mistake and choose L, resulting in a payoff of 1. When a perfect information game is solved using standard backwards induction (minimax evalua-

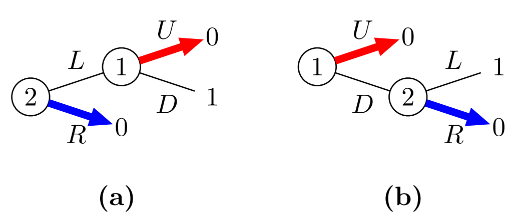

Figure 1: **(a)** Not subgame perfect. **(b)** Not admissible.

*Department of computer science, University of Aarhus, Denmark. Email {bromille,trold}@daimi.au.dk. Research supported by *Center for Algorithmic Game Theory*, funded by the Carlsberg Foundation.

tion), mistakes of type (a) are automatically avoided and it is straightforward to break ties in the minimax evaluation so as to avoid mistakes of type (b). However, if the game is solved using the Koller-Megiddo-von Stengel algorithm, the "bad" equilibria may easily be given as output, and if the game has imperfect information, straightforward techniques for avoiding them do not apply.

Fortunately, in game theory, *refinements* of Nash equilibrium were defined with exactly the purpose of eliminating "insensible" equilibria. For a comprehensive account, see the monograph of van Damme [5]. Using terminology from this theory, the equilibrium in (a) is not *subgame perfect*, since its restriction to the subgame starting with the position where Player 1 gets to move is not a Nash equilibrium. The equilibrium in (b) *is* subgame perfect, but it is not an equilibrium using *admissible strategies*, as the strategy it prescribes for Player 1 is weakly dominated by his other possible strategy. An attractive equilibrium refinement for extensive form games is *quasi-perfect equilibrium* which was introduced by van Damme [4]. A quasi-perfect equilibrium is guaranteed to be using admissible strategies and is also guaranteed to be *sequential*, a non-trivial refinement of subgame perfection for imperfect information games due to Kreps and Wilson [12]. In previous work [16, 17], we showed how to modify the Koller-Megiddo-von Stengel linear programs so that a quasi-perfect equilibrium is computed. This eliminates the undesirable behavior in all the examples pointed out by Koller and Pfeffer.

However, while insisting on a quasi-perfect equilibrium eliminates most of the cases of "returning gifts" in the computed equilibria, there are still games that are not solved in a satisfactory way. Here, we show an example of an equilibrium of a fairly natural extensive-form game we call *Matching Pennies on Christmas Morning* and first presented in [15]. The game is as follows. In the standard Matching Pennies game, Player 2 (Bob) hides a penny and Player 1 (Alice) has to guess if it is heads or tails up. If she guesses correctly, she gets the penny. If played on Christmas morning, we add a *gift option*: After Player 2 has hidden his penny but before Player 1 guesses, Player 2 may choose to publicly give Player 1 a gift of one penny, in addition to the one Player 1 will get if she guesses correctly. The extensive form of this game as well as the pair of maximin/minimax behavioral strategies are given in Figure 2. We see that if Player 1 does not receive a gift, the strategy indicated suggests that she randomizes her guess and guesses *heads* with probability $\frac{1}{2}$ and *tails* with probability $\frac{1}{2}$. This is indeed the strategy we expect to see. On the other hand, if Player 1 *does* receive a gift, the strategy indicated suggests that she guesses *heads* with

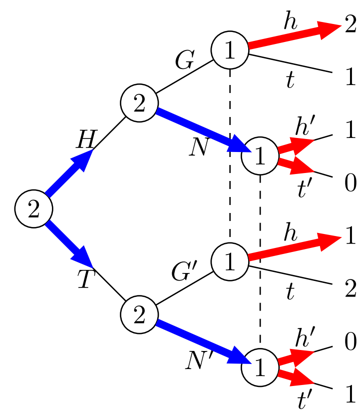

Figure 2: *Matching Pennies on Christmas Morning* - "bad" equilibrium (lengths of arrows indicate behavior probabilities)

probability 1. This does not seem sensible. Indeed, if she had randomized her guess, as in the "no-gift" scenario, her worst case conditional expected payoff, conditioned by the observed fact that she received the gift, would be guaranteed to be at least a penny and a half. For the strategy computed, this worst case conditional expected payoff is only a penny. The "bad" equilibrium is quasi-perfect, and all previously described variants of the Koller-Megiddo-von Stengel algorithm (including our previous work [16, 17]) may give it as output. In particular, the implementation of the algorithm included in the game theory software tool Gambit [14] outputs exactly this equilibrium. Observing that any *basic* solution to the linear program devised by Koller, Megiddo and von Stengel prescribes deterministic play after getting the gift, one would in fact have been quite surprised had one seen randomized play.

The only standard equilibrium refinement we are aware of that only permits the "sensible" equilibrium of Matching Pennies on Christmas Morning is the notion of *proper equilibrium* of Myerson [18]. An equilibrium in mixed strategies for a bimatrix game is said to be proper if it is a limit point of a sequence of ϵ-proper completely mixed strategy profiles for $\epsilon \to 0+$. Here, a strategy profile is said to be completely mixed if it prescribes strictly positive probability to every pure strategy. It is said to be ϵ-proper if the following property is satisfied: If pure strategy x_i is a better reply than pure strategy x_j against the mixed strategy that the profile prescribes to the other player, we have $p(x_j) \leq \epsilon p(x_i)$ where $p(x_k)$ is the probability prescribed to pure strategy x_k. An equilibrium in behavior strategies for an extensive form two-player game with perfect recall is defined to be *normal form proper* if it is behaviorally equivalent to a proper equilibrium of the corresponding normal form game. Using fixed point arguments, Myerson showed the existence of a normal form proper equilibrium for any such game. It is easy to see that a normal form proper equilibrium is also a Nash equilibrium.

The intention of the definition of proper equilibrium is to capture the behavior of players that assume that their opponents sometimes make mistakes, but with negligible probability and *in a rational manner*: Players assume that their opponents play more costly suboptimal strategies with significantly smaller probability than less costly suboptimal strategies, but that any suboptimal strategy is played with infinitesimally small probability. It can be seen that the unique normal form proper equilibrium of Matching Pennies on Christmas Morning is the one where Player 1 guesses uniformly at random, also after having received a gift. The intuitive reason is that even after observing that Player 2 has made a mistake by giving the gift, Player 1 assumes that Player 2 in all other respects behaves rationally. In particular, she assumes that he will exploit any bias in the guess she makes, and hence she has to avoid any such bias.

Kohlberg and Mertens [9] and van Damme [4, 5] established a number of attractive properties of normal form proper equilibrium, yielding additional motivation for computing it. First, any normal form proper equilibrium in behavior plans (plans are "reduced" strategies that do not specify actions at irrelevant information sets, see Section 2) is also a quasi-perfect equilibrium in behavior plans [9, 4], so examples where the "sensible" solutions are guaranteed by quasi-perfection (see [16] for non-trivial such examples) are still solved in a satisfactory way when a normal form proper equilibrium is computed. Also, for the case of a zero-sum game, van Damme [5, Theorem 3.5.5] has shown that the set of proper equilibria is a Cartesian product $D \times D'$, where D is a polytope of mixed strategies for Player 1 and D' is a polytope of mixed strategies for Player 2. This nice fact makes it possible to extend the properness terminology from strategy profiles to strategies: We define a proper mixed *strategy* for Player 1 (resp., 2) to be an element of D (resp., D') and a proper behavior strategy to be a behavior strategy behaviorally equivalent to a proper mixed strategy. Note that this is completely analogous to the case of Nash equilibria which for the case of a zero-sum game is the Cartesian product of the maximin strategies of Player 1 and the minimax strategies of Player 2, by von Neuman's min-max theorem.

The above discussion motivates the computation of proper strategies when prescriptive behavior in an extensive form zero-sum game is desired. In the present paper, we show how to extend the approach of Koller, Megiddo and von Stengel and do such a computation in polynomial time. First, we study the case of *imperfect information* games with perfect recall and show that the normal form proper equilibria of such games can be completely characterized by a procedure involving an iteratively defined sequence of linear programs derived from the linear programs for maximin behavior plans in sequence form described by Koller, Megiddo and von Stengel. Each linear program in the sequence has a number of variables and constraints which is at most the size of the game tree and the number of programs in the sequence is also at most the size of the game tree. This establishes our first main theorem:

THEOREM 1.1. *A normal form proper equilibrium in behavior plans for a given extensive form two-player zero-sum game of imperfect information but perfect recall can be found in polynomial time in the size of the game tree.*

The sequence of linear programs we construct is analogous to the sequence of linear programs arising in *Dresher's procedure* [7] established by van Damme [5] as characterizing the proper equilibria of a matrix game and our proof of correctness is based on similar ideas as van Damme's proof of this fact. Thus, our algorithm may be seen as a sequence form version of Dresher's procedure. However, the programs we devise are not in general equivalent to the linear programs arising in Dresher's procedure for the corresponding normal form game, and indeed, for some games our sequence of programs is exponentially shorter than the sequence of programs arising in Dresher's procedure.

Our main motivation for computing proper equilibria is computing sensible prescriptive behavior in imperfect information games. But, somewhat surprisingly, it turns out that normal form proper equilibrium is an interesting solution concept even for the case of *perfect* information games. As an example, consider the game given in Figure 3. The value of the game is 0 and Player 1 is guaranteed to obtain this value no matter what she does. However, if she chooses action U and her opponent makes a mistake, she will receive a payoff of 1. On the other hand, if she chooses action D and her opponent makes a mistake, she will receive a payoff

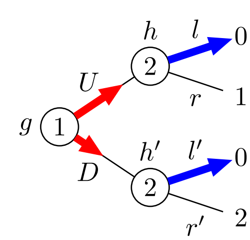

Figure 3: *Up or Down?*

	ll'	lr'	rl'	rr'	
U	0	0	1	1	2/3
D	0	2	0	2	1/3
	$1-3\epsilon-\epsilon^2$	$\epsilon-\epsilon^2$	2ϵ	$2\epsilon^2$	

Figure 4: The normal form of *Up or Down?* and an ϵ-proper strategy profile.

of 2. In the unique normal form proper equilibrium for this game, Player 1 chooses U with probability $\frac{2}{3}$ and D with probability $\frac{1}{3}$. An intuitive justification for this behavior strategy is as follows. Player 1 must imagine being up against a Player 2 that cannot avoid sometimes making mistakes - otherwise her choice is irrelevant. On the other hand, she should assume that Player 2 is still a rational player who makes an effort to avoid making mistakes. In particular, she should assume that Player 2 makes a strategic decision about whether to train to avoid making the mistake in position h or to train to avoid making the mistake in position h', prior to playing the game. Thus, the strategy of Player 1 should not be pure. In particular, if she chooses D with probability 1 (as is surely tempting), Player 2 may respond by concentrating all his efforts to avoid making mistakes in h'. Then, Player 1 will not get her "fair share" of payoff from Player 2's mistakes. This reasoning is somewhat analogous to the heuristic reasoning suggesting why it is a good idea for expert chess players to randomize which opening to select against other expert players, despite the fact that chess is a perfect information game.

To see more precisely why the behavior strategy suggested is the unique proper equilibrium, we can look at the normal form of the game. This is the matrix game with the matrix given in Figure 4. A (3ϵ)-proper strategy profile is given (for sufficiently small $\epsilon > 0$) by the polynomials in ϵ stated in the last row and column of the matrix. The limit as $\epsilon \to 0$ gives the desired proper strategy for the row player. It is also relatively easy to see that no other mix of her two strategies can give rise to an ϵ-proper strategy profile. Hence, the stated proper strategy is the unique one.

Of course, for larger games we cannot afford the conversion of extensive to normal form. This motivates our second main result:

Theorem 1.2. *A normal form proper equilibrium in behavior strategies for a given extensive form two-player zero-sum game of perfect information can be found in linear time in the size of the tree.*

The procedure we exhibit is a *backwards induction* procedure, refining the standard backwards induction procedure (minimax evaluation) for computing a subgame perfect equilibrium. As a curious example, applying the procedure to *tic-tac-toe* one finds that in any normal form proper equilibrium of this game, the game is opened by selecting the middle square (as one of nine opening moves) with probability $\frac{1}{13}$.

1.1 Related Research and Open Problems In his monograph, van Damme established that a procedure due to Dresher [7] characterizes the proper equilibria of a matrix game. One may apply this procedure to an extensive form game by first converting the game to normal form (incurring an exponential blowup in the size of the representation) and then applying the procedure. As described by van Damme (and Dresher), the procedure uses exponential time as it involves explicitly enumerating the vertices of a polytope defined by a set of inequalities, so this would yield a doubly exponential time procedure. In previous work [15], we showed how to relatively easily modify the procedure so that it becomes polynomial time, yielding a polynomial time procedure for computing a proper equilibrium of a *matrix* game and hence a singly exponential time procedure for computing a normal form proper equilibrium for a two-player zero-sum extensive form game with perfect recall. The procedure we present in this paper is much more subtle. It subsumes the procedure presented in [15] in the following sense: If a matrix game is represented as an extensive form game of depth two with one information set for each player and the procedure of the present paper is applied to this representation, we get essentially the same sequence of linear programs as when applying the procedure of [15] to the matrix game.

Computing a proper equilibrium of a *bimatrix* game remains an elusive problem (whose solution is a prerequisite for even considering computing a normal form proper equilibrium for a two-player extensive form non-zero sum game). Yamamoto [20] presents a numerical procedure for performing such a computation. It involves solving certain differential equations numerically and it is not clear under which circumstances it can be formally guaranteed to compute (or converge to) a proper equilibrium. One should note that since proper equilibrium refines Nash equilibrium, it is PPAD-hard to compute it, even for the bimatrix case [6, 3]. Given this, we find the following problem very interesting.

Open Problem 1. *Is computing a proper equilibrium of a bimatrix game in PPAD?*

Another problem concerns the computation of ϵ-proper strategy profiles. We do not know how to perform this computation, even for matrix games. To be precise, we would like an algorithm which given a matrix game outputs a parameterized strategy profile with each probability being a formal polynomial in ϵ and so that by inserting a sufficiently small concrete value of $\epsilon > 0$ in the polynomials, we get an ϵ-proper strategy profile (as in Figure 4). The existence of such a polynomial ϵ-proper strategy profile of any game is not obvious but follows from results of Blume et al. [2]. Note that the constant terms of the polynomials constitute a proper equilibrium, so we do know how to compute at least these terms, but not the rest of the polynomials.

OPEN PROBLEM 2. *Can an ϵ-proper strategy profile (as a system of formal polynomials in ϵ) of a given matrix game be computed in polynomial time?*

1.2 Organization of Paper We present the necessary background in Section 2, followed by our procedure for imperfect information games in Section 3 and our procedure for perfect information games in Section 4. Due to lack of space, the proofs of correctness are omitted from this version of the paper. They can be found in the version available at the home pages of the authors.

2 Preliminaries

We describe the relevant concepts of extensive-form games and in particular the results of von Stengel [19] and Koller, Megiddo and von Stengel [10] that we base our algorithms upon.

A two-player, extensive-form zero-sum game G with imperfect information but perfect recall is given by a finite tree with payoffs at the leaves, information sets partitioning nodes of the tree and nodes of chance being allowed. Actions of a player are denoted by labels on edges of the tree. Perfect recall means that all nodes in an information set belonging to a player share the sequence of actions and information sets of that player that are visited on the path from the root to the nodes. For convenience, we assume that actions taken in different information sets have different names. In particular, from the name of an action, we can deduce its information set. Also, by perfect recall, from the name c of an action, we can deduce the sequence of actions σ taken before c by the same player.

A *pure strategy* s for a player is a *set* of designated actions, containing exactly one action of each information set belonging to him. A *mixed strategy* for a player is a probability distribution over pure strategies. Since there are exponentially many pure strategies as a function of the number of information sets, it is usually not feasible to explicitly represent mixed strategies. A more compact notion is the notion of a *behavior strategy*. A behavior strategy is simply an assignment of probabilities to actions, and therefore an object of size comparable to the description of the game itself. Kuhn [13] showed that for games of perfect recall, for any mixed strategy p, there is a behavior strategy π that is *behaviorally equivalent*. That is, a player playing by p will generate exactly the same distribution of plays against any opponent as a player playing by π. Thus, we can compactly represent mixed strategies by behavior strategies. An even more compact notion is the notion of a *behavior plan*. This is an assignment of probabilities to actions of those information sets *that may be reached with non-zero probability if the plan is played* against some strategy of the opponent. These information sets are called *relevant* for the plan. No meaningful behavior is assigned to information sets that are guaranteed not to be reached if the plan is played, no matter what the opponent does. These information sets are called *irrelevant* for the plan. Behavior strategies and behavior plans are clearly behaviorally equivalent.

An important insight of von Stengel [19] and Koller, Megiddo and von Stengel is that behavior plans for games of perfect recall are for computational purposes often better represented in *sequence form*, which we describe next. Given a behavior strategy for one of the players, the *realization weight* of a sequence σ of actions made by that player is the *product* of behavior probabilities of the actions in the sequence. A *realization plan* for a player is a vector of realization weights indexed by his possible sequences of actions. Note that a behavior probability is the ratio between two realization weights. If this ratio is $0/0$, the realization weights do not define a behavior probability. Still, the map between behavior *plans* and realization plans is a bijection, so given a realization plan, we may talk about the corresponding behavior plan and vice versa. Note that a pure strategy corresponds to a realization plan that is a 0/1-vector. For a game G of perfect recall, the statement that a vector is a valid realization plan can be expressed by a non-negativity constraint and a system of linear equations indexed by information sets. Here, for convenience, we include an artificial "pre-game" information set called 0 and belonging to both players. We let

$$Ex = e, x \geq 0 \qquad (2.1)$$

denote the constraints expressing that x is a valid realization plan for Player 1 and we let

$$Fy = f, y \geq 0 \qquad (2.2)$$

be the constraints expressing that y is a valid realization plan for Player 2. Here, E and F are matrices containing entries from $\{-1, 0, 1\}$. The rows of E (F) are indexed by information sets of Player 1 (Player 2) and the vector

e (f) is the vector with a 1 in the entry indexed by the pre-game information set 0 and with 0 in every other entry.

The key to the computational usefulness of sequence form is the following fact: If Player 1 plays by realization plan x and Player 2 plays by realization plan y, then the expected payoff for Player 1 is given by a *bilinear form* $x^\top Ay$, where A is easily computed from the description of the game. Koller, Megiddo and von Stengel used this to show that the set of maximin realization plans for Player 1 is given by the x-parts of optimal solutions to the following linear program.

$$\begin{array}{rl} \max\limits_{x,q} & f^\top q \\ \text{s.t.} & -A^\top x + F^\top q \leq 0 \\ & Ex = e \\ & x \geq \mathbf{0} \end{array} \tag{2.3}$$

The value of the objective function in an optimal solution is the value of the game. Note that $f^\top q$ is simply q_0 where 0 is the artificial pre-game information set. The inequalities in the system $-A^\top x + F^\top q \leq 0$ are indexed by action sequences of Player 2. For understanding how the program works, it is very useful to know the following interpretation of the variable q_h: Player 2 can guarantee that the contribution to the expected payoff from plays where he finds himself in information set h is at most q_h. We may illustrate this by exhibiting the program for Player 1 in the game *Up or Down?* from Figure 3. The matrices defining the sequence form of the game are given below. Most zero-entries are omitted for readability.

$$A = \begin{array}{c} \begin{array}{ccccc} \lambda & l & r & l' & r' \end{array} \\ \left[\begin{array}{ccccc} & & & & \\ & 0 & 1 & & \\ & & & 0 & 2 \end{array}\right] \end{array} \begin{array}{l} \lambda \\ U \\ D \end{array}$$

$$E = \begin{array}{c} \begin{array}{ccc} \lambda & U & D \end{array} \\ \left[\begin{array}{ccc} 1 & & \\ -1 & 1 & 1 \end{array}\right] \end{array} \begin{array}{l} 0 \\ g \end{array} \qquad F = \begin{array}{c} \begin{array}{ccccc} \lambda & l & r & l' & r' \end{array} \\ \left[\begin{array}{ccccc} 1 & & & & \\ -1 & 1 & 1 & & \\ -1 & & & 1 & 1 \end{array}\right] \end{array} \begin{array}{l} 0 \\ h \\ h' \end{array}$$

Here, λ is the empty sequence. Now, the linear program (2.3) may be written as follows:

$$\begin{array}{rl} \max\limits_{x,q} & q_0 \\ \text{s.t.} & q_0 \leq q_h + q_{h'} \\ & q_h \leq 0 \\ & q_h \leq x_U \\ & q_{h'} \leq 0 \\ & q_{h'} \leq 2x_D \\ & x_\lambda = 1 \\ & x_U + x_D = x_\lambda \\ & x_\lambda, x_U, x_D \geq 0 \end{array}$$

Note that any optimal solution has $q_h = q_{h'} = q_0 = 0$ while any realization plan is an optimal choice of x. The optimal value of the objective function is 0 as is indeed the value of the game. To anticipate our procedure in the next section, we might however already now note some differences between the solution with $x_U = 1, x_D = 0$ and the solution with $x_U = 0, x_D = 1$: The former solution has *no slack* in the inequality $q_{h'} \leq 2x_D$ but obtains the maximum possible slack in the inequality $q_h \leq x_U$ while the situation is the opposite for the latter solution. This corresponds to the fact that the former solution does not exploit opponents making a mistake in information set h' while the latter solution does not exploit opponents making a mistake in information set h. As we explained in the introduction, a proper equilibrium has to exploit both possible mistakes. *Balancing* the slack obtained by the solution in the various inequalities of the system $-A^\top x + F^\top q \leq 0$ turns out to be the key to computing a proper equilibrium.

3 The procedure for imperfect information games

Let G be a two-player zero-sum extensive form game with perfect recall played between Player 1, trying to maximize payoff and Player 2, trying to minimize payoff. Our construction uses the *sequence form* of G and is based on the linear programming characterization of Nash equilibria in realization plans due to Koller, Megiddo and von Stengel [10], described in the previous section. In particular, G is given by a payoff matrix A, and realization plan constraint matrices E and F for Player 1 and Player 2 respectively. We define a series of linear programs where the coefficients of each linear program depend on the solutions of previous linear programs. We group the linear programs in pairs, denoting a pair as a round, the first being round 0. In round 0, we first consider the following linear program $P^{(0)}$ which is the linear program devised by Koller, Megiddo and von Stengel for characterizing the maximin realization plans for Player 1 and computing the value of the game.

$$\begin{array}{rl} \multicolumn{2}{c}{\underline{P^{(0)}}:} \\ \max\limits_{x,q} & f^\top q \\ \text{s.t.} & -A^\top x + F^\top q \leq 0 \\ & Ex = e \\ & x \geq \mathbf{0} \end{array} \tag{3.4}$$

The vector variable x is indexed by action sequences of Player 1 and in the optimum solution describes a maximin realization plan for Player 1. The vector q is indexed by information sets of Player 2. We let $v^{(0)}$

be the value of the optimal solution. This is the value of the game, by the result of Koller, Megiddo and von Stengel. Next, we consider the following program $Q^{(0)}$.

$$
\begin{array}{llr}
\underline{Q^{(0)}}: & & \\
\max\limits_{x,q,u,s} & \mathbf{1}^\top u & \\
\text{s.t.} & -A^\top x + F^\top q + u & \leq \mathbf{0} \\
& Ex \qquad\qquad - es & = \mathbf{0} \\
& f^\top q \qquad - v^{(0)} s & = 0 \\
& & \mathbf{0} \leq u \leq \mathbf{1} \\
& & x \geq \mathbf{0} \\
& & s \geq 1
\end{array}
\tag{3.5}
$$

The variable s is scalar. Informally, feasible solutions of $Q^{(0)}$ are feasible solutions of $P^{(0)}$ *scaled* by s. The vector variable u is indexed by action sequences of Player 2. All optimal solutions to (3.5) have the same value of this u-vector, which always takes the form of a 0/1-vector (the possibility of setting the scaling variable s to an arbitrarily large value ensures this). The 1-entries in this 0/1-vector determine those inequalities of $P^{(0)}$ that may have *slack* in an optimal solution to $P^{(0)}$. Intuitively, they identify certain action sequences of Player 2 as *containing* mistakes. Let $\tilde{m}^{(1)}$ be this optimal u. We let $\tilde{M}^{(1)}$ be the set of sequences on which $\tilde{m}^{(1)}$ is 1. Next, we identify a set of actions as *being* mistakes. These are the actions that are the final actions in sequences containing mistakes with the additional condition that removing the action from the sequence results in a sequence not containing any mistakes. Formally, we let $m^{(1)}$ be defined by

$$
m^{(1)}_{\sigma c} = \begin{cases} 1 & \text{if } \tilde{m}^{(1)}_{\sigma} = 0 \wedge \tilde{m}^{(1)}_{\sigma c} = 1 \\ 0 & \text{otherwise} \end{cases}
\tag{3.6}
$$

We let $M^{(1)}$ be the set of actions c for which $m^{(1)}$ is 1 on the sequence ending in c. We denote this set as the *mistakes of order* 1 of Player 2. If $m^{(1)} \neq \mathbf{0}$, it defines the next round of linear programs.

Assuming we have already defined rounds $0, \ldots, k-1$, the k'th round looks as follows. We define the linear program $P^{(k)}$:

$$
\begin{array}{llr}
\underline{P^{(k)}}: & & \\
\max\limits_{x,q,t} & t & \\
\text{s.t.} & -A^\top x + F^\top q + m^{(k)} t & \leq -\sum\limits_{0<i<k} m^{(i)} v^{(i)} \\
& Ex & = e \\
& f^\top q & = v^{(0)} \\
& & x \geq \mathbf{0} \\
& & t \geq 0
\end{array}
\tag{3.7}
$$

The variable t is scalar. We let $v^{(k)}$ be the value of t in an optimal solution of (3.7). Informally, $v^{(k)}$ is the *maximin slack* achievable in the inequalities indexed by $M^{(k)}$ (the min being over the inequalities). Note that $P^{(0)}$ is special and does not have the same format as $P^{(k)}$ for $k \geq 1$. Still, it is clear that the feasible solutions of $P^{(1)}$ are exactly the optimal solutions of $P^{(0)}$. In general, we will have that the feasible solutions of $P^{(k+1)}$ are the optimal solutions of $P^{(k)}$. Next, we consider the following linear program $Q^{(k)}$.

$$
\begin{array}{llr}
\underline{Q^{(k)}}: & & \\
\max\limits_{x,q,u,s} & \mathbf{1}^\top u & \\
\text{s.t.} & -A^\top x + F^\top q + u + \sum\limits_{0<i\leq k} m^{(i)} v^{(i)} s & \leq \mathbf{0} \\
& Ex \qquad\qquad - es & = \mathbf{0} \\
& f^\top q \qquad - v^{(0)} s & = 0 \\
& & \mathbf{0} \leq u \leq \mathbf{1} \\
& & x \geq \mathbf{0} \\
& & s \geq 1
\end{array}
\tag{3.8}
$$

As was the case for $Q^{(0)}$, all optimal solutions to (3.8) have the same value of the u-vector, which always takes the form of a 0/1-vector. The 1-entries in this 0/1-vector determine the inequalities of $P^{(k)}$ that may have slack in an optimal solution to $P^{(k)}$. Let $\tilde{m}^{(k+1)}$ be this optimal u. We let $\tilde{M}^{(k+1)}$ be the set of sequences on which $\tilde{m}^{(k+1)}$ is 1. Let $m^{(k+1)}$ be defined by

$$
m^{(k+1)}_{\sigma c} = \begin{cases} 1 & \text{if } \tilde{m}^{(k+1)}_{\sigma} = 0 \wedge \tilde{m}^{(k+1)}_{\sigma c} = 1 \\ 0 & \text{otherwise} \end{cases}
\tag{3.9}
$$

We let $M^{(k+1)}$ be the set of actions c for which $m^{(k+1)}$ is 1 on the sequence ending in c. We denote this set as the *mistakes of order* $k+1$ of Player 2. With this terminology, an informal interpretation of the meaning of the constant $v^{(k)}$ is the *cost* for Player 2 of each mistake of order k that he includes in his chosen strategy. This interpretation is formalized in the correctness proof for the procedure. The sets $M^{(k)}$ are not necessarily disjoint, i.e., a specific action may be a mistake of several different orders. It can be seen that the set of orders of a given mistake is an interval $[i, i+1, \ldots, j]$. This completes the description of the k'th round. If $m^{(k+1)} \neq \mathbf{0}$, it defines the next round of linear programs.

In some round $k = K$, an optimal solution to (3.8) has $u = \mathbf{0}$ and the procedure is terminated. We argue in the full version of the paper that this happens after at most a number of rounds equal to the number of actions of Player 2 in the game tree. Let D be the x-parts of the set of optimal solutions to $P^{(K)}$. This is a set of realization plans for Player 1. Interchanging the role of Player 1 and Player 2 and negating the payoff matrix, we carry out the entire procedure again. Let D' be the resulting set of realization plans for Player 2. We can now state the main theorem of the present section.

THEOREM 3.1. *$D \times D'$ is the set of normal form proper equilibria of G in realization plans.*

Since the number of rounds is at most the number of actions, standard arguments imply that the number of bits needed to describe the coefficients of the linear programs remains polynomially bounded throughout the iteration. Thus the theorem implies that we have a polynomial time algorithm for computing a normal form proper equilibrium in realization plans (and hence behavior plans) for a given two-player, zero-sum extensive form game of imperfect information but perfect recall. That is, we have established Theorem 1.1 of the introduction. However, one should notice that Theorem 3.1 allows us not only to compute *one* normal form proper equilibrium; it allows us to exhibit a linear program characterizing all of them.

To give a better intuition of how the algorithm works, we show the sequence of linear programs arising when finding the proper strategies for Player 1 in the game *Up or Down?* from Figure 3. The first linear program to solve is $P^{(0)}$, given below. This is the original Koller-Megiddo-von Stengel program that was already described in the previous section.

$$
\begin{array}{lr}
\underline{P^{(0)}}: & \\
\max_{x,q} & q_0 \\
\text{s.t.} & q_0 \leq q_h + q_{h'} \\
& q_h \leq 0 \\
& q_h \leq x_U \\
& q_{h'} \leq 0 \\
& q_{h'} \leq 2x_D \\
& x_\lambda = 1 \\
& x_U + x_D = x_\lambda \\
& x_\lambda, x_U, x_D \geq 0
\end{array}
$$

As already mentioned, any optimal solution has $q_h = q_{h'} = q_0 = 0$ while any realization plan is an optimal choice of x. The optimal value of the objective function is $v^{(0)} = 0$. This value is used in the construction of $Q^{(0)}$, given below.

$$
\begin{array}{lr}
\underline{Q^{(0)}}: & \\
\max_{x,q,u,s} & u_\lambda + u_l + u_r + u_{l'} + u_{r'} \\
\text{s.t.} & q_0 + u_\lambda \leq q_h + q_{h'} \\
& q_h + u_l \leq 0 \\
& q_h + u_r \leq x_U \\
& q_{h'} + u_{l'} \leq 0 \\
& q_{h'} + u_{r'} \leq 2x_D \\
& x_\lambda = s \\
& x_U + x_D = x_\lambda \\
& q_0 = 0 \\
& 0 \leq u_\lambda, u_l, u_r, u_{l'}, u_{r'} \leq 1 \\
& x_\lambda, x_U, x_D \geq 0 \\
& s \geq 1
\end{array}
$$

All optimal solutions to $Q^{(0)}$ have $u_{\{\lambda,l,l'\}} = \mathbf{0}$ and $u_{\{r,r'\}} = \mathbf{1}$, indicating that r and r' are the two possible mistakes of order 1 that Player 2 can make. This optimal u defines the vector $m^{(1)}$ of mistakes, which is used in the construction of $P^{(1)}$, given below.

$$
\begin{array}{lr}
\underline{P^{(1)}}: & \\
\max_{x,q} & t \\
\text{s.t.} & q_0 \leq q_h + q_{h'} \\
& q_h \leq 0 \\
& q_h + t \leq x_U \\
& q_{h'} \leq 0 \\
& q_{h'} + t \leq 2x_D \\
& x_\lambda = 1 \\
& x_U + x_D = x_\lambda \\
& x_\lambda, x_U, x_D, t \geq 0 \\
& q_0 = 0 \\
& x_\lambda, x_U, x_D, t \geq 0
\end{array}
$$

This program maximizes the minimum slack in the inequalities corresponding to the mistakes found. In this case, the maximin slack achievable is $t = \frac{2}{3}$, achieved by setting $x_U = \frac{2}{3}$ and $x_D = \frac{1}{3}$. Intuitively, it is the cost for Player 2 of making either of his two possible mistakes. Inserting $v^{(1)} = \frac{2}{3}$ into (3.8), we obtain $Q^{(1)}$ given below.

$$
\begin{array}{lr}
\underline{Q^{(1)}}: & \\
\max_{x,q,u,s} & u_\lambda + u_l + u_r + u_{l'} + u_{r'} \\
\text{s.t.} & q_0 + u_\lambda \leq q_h + q_{h'} \\
& q_h + u_l \leq 0 \\
& q_h + u_r \leq x_U - \frac{2}{3}s \\
& q_{h'} + u_{l'} \leq 0 \\
& q_{h'} + u_{r'} \leq 2x_D - \frac{2}{3}s \\
& x_\lambda = s \\
& x_U + x_D = x_\lambda \\
& q_0 = 0 \\
& 0 \leq u_\lambda, u_l, u_r, u_{l'}, u_{r'} \leq 1 \\
& x_\lambda, x_U, x_D \geq 0 \\
& s \geq 1
\end{array}
$$

All feasible solutions to $Q^{(1)}$ have $u = 0$, so there are no mistakes of order 2 for Player 1 to exploit. By Theorem 3.1, any optimal x to $P^{(1)}$ is therefore a proper strategy. The only one in this case is $x_U = \frac{2}{3}$, $x_D = \frac{1}{3}$.

Due to lack of space the proof of Theorem 3.1 is not included in this version of the paper. The main idea is similar to the proof of van Damme's characterization of the proper equilibria of a matrix game [5, Theorem 3.5.5]. We prove that a proper strategy will be an optimal solution to all the linear programs $P^{(k)}$ and we prove that any two optimal solutions to the final program $P^{(K)}$ are payoff-equivalent. Theorem 3.1 then follows easily from these two facts.

4 The procedure for perfect information games

Before we present our procedure for perfect information games, we want to address a subtle point. Our result for imperfect information games characterizes the normal form proper equilibria in realization plans which are equivalent to behavior *plans*. That is, no meaningful behavior is described in *irrelevant* information sets for the plans (See Section 2 for definitions). This is an inherent property of the notion of normal form proper equilibrium. However, it can be argued that it is sometimes desirable to prescribe meaningful behavior at irrelevant information sets. An example is the computation of prescriptive strategies for game playing software; one could imagine the software being used for advisory purposes with the user having the opportunity of ignoring the advice. Then, the user might still want meaningful advice after having ignored a particular piece of advice.

A slightly more restrictive concept than normal form properness that also provides meaningful behavior in irrelevant information sets was suggested by van Damme [4]. He calls an equilibrium in behavior strategies of an extensive form game *induced* by a normal form proper equilibrium if it is a limit of behavior strategies given by a sequence of ϵ-proper equilibria of the corresponding normal form. We shall adopt the slightly more convenient terminology *induced normal form proper equilibrium* for such an equilibrium. Note that while the induced normal form proper equilibrium is the *limit* of the behavior strategies *behaviorally equivalent* to the sequence of the ϵ-proper strategy profiles, the normal form proper equilibrium is the behavior strategy *behaviorally equivalent* to the *limit* of the sequence. It can be seen that the normal form proper equilibria are simply those equilibria that can be obtained by taking an induced normal form proper equilibrium and replacing the behavior in irrelevant information sets with arbitrary behavior. In particular, if we consider equilibria in behavior plans (as in the previous section) rather than behavior strategies the two notions coincide. But for perfect information games, we are able to characterize and compute the induced normal form proper equilibria in behavior *strategies*, rather than plans. It is an interesting open problem to do this for imperfect information games.

Let G be a perfect information zero-sum game played between Player 1, trying to maximize payoff and Player 2, trying to minimize payoff. The game is given by a game tree with payoffs in leaves and each internal node belonging to either Player 1, Player 2 or Chance. For each node i in the tree we associate three number $\underline{v}_i \leq v_i \leq \overline{v}_i$. The number v_i is the usual minimax value of the node and may be computed by standard backwards induction. The values $\underline{v}_i$ and $\overline{v}_i$ can be informally seen as pessimistic and optimistic estimates of the expected outcome of the game from the point of Player 1, taking the possibility of mistakes being made by either player into account.

For a leaf with payoff p we let $\overline{v}_i = v_i = \underline{v}_i = p$. For an internal node i, we denote the set of immediate successors of i by $S(i)$ and define $\underline{v}_i, \overline{v}_i$ inductively as follows.

If i is a node belonging to Player 1, we let $V_i = (\cup_{j \in S(i)} \{v_j, \underline{v}_j\}) \setminus \{v_i\}$, i.e., the set of all values and all pessimistic estimates of all immediate successors of i, *except* the value of i itself. Then we let

$$\underline{v}_i = \begin{cases} \max(V_i) & \text{if } V_i \neq \emptyset, \\ v_i & \text{otherwise.} \end{cases} \tag{4.10}$$

Also, for a node i belonging to Player 1, we let $I_i = \{j \in S(i) | v_j = v_i \wedge \overline{v}_j > v_j\}$, i.e., the set of immediate successors of i having the same value as i and a non-trivial optimistic value, and let

$$\overline{v}_i = v_i + \begin{cases} \frac{1}{\sum_{j \in I_i} (\overline{v}_j - v_j)^{-1}} & \text{if } I_i \neq \emptyset, \\ 0 & \text{otherwise.} \end{cases} \tag{4.11}$$

Similarly, if i is a node belonging to Player 2, we let $V_i = (\cup_{j \in S(i)} \{v_j, \overline{v}_j\}) \setminus \{v_i\}$, and let

$$\overline{v}_i = \begin{cases} \min(V_i) & \text{if } V_i \neq \emptyset, \\ v_i & \text{otherwise.} \end{cases} \tag{4.12}$$

Also, for node i belonging to Player 2, we let $I_i = \{j \in S(i) | v_j = v_i \wedge \underline{v}_j < v_j\}$ and let

$$\underline{v}_i = v_i - \begin{cases} \frac{1}{\sum_{j \in I_i} (v_j - \underline{v}_j)^{-1}} & \text{if } I_i \neq \emptyset, \\ 0 & \text{otherwise.} \end{cases} \tag{4.13}$$

If i is a node belonging to Chance and $j \in S(i)$ is chosen by Chance with probability α_j, we let $R_i = \{j \in S(i) | \underline{v}_j < v_j\}$ and $R'_i = \{j \in S(i) | \overline{v}_j > v_j\}$ and let

$$\underline{v}_i = v_i - \min_{j \in R_i} \alpha_j (v_j - \underline{v}_j) \tag{4.14}$$

$$\overline{v}_i = v_i + \min_{j \in R'_i} \alpha_j (\overline{v}_j - v_j) \tag{4.15}$$

Our characterization of induced proper equilibria is then the following:

THEOREM 4.1. *A behavior strategy profile ρ for a two-player, zero-sum, perfect information game G is an induced normal form proper equilibrium of G* if and only if *the following three conditions all hold:*

1. *For all nodes i and immediate successors j, ρ assigns non-zero behavior probability to j only if $v_i = v_j$.*

2. *For all nodes i belonging to Player 1 for which $I_i \neq \emptyset$, ρ assigns behavior probability* exactly $\frac{(\overline{v}_j - v_j)^{-1}}{\sum_{j \in I_i} (\overline{v}_j - v_j)^{-1}}$ *to each $j \in I_i$.*

3. *For all nodes i belonging to Player 2 for which $I_i \neq \emptyset$, ρ assigns behavior probability* exactly $\frac{(v_j - \underline{v}_j)^{-1}}{\sum_{j \in I_i} (v_j - \underline{v}_j)^{-1}}$ *to each $j \in I_i$.*

Note that the first condition is exactly the condition of being a subgame perfect equilibrium, while the other conditions further restrict the behavior probabilities of choices made in nodes i for which $I_i \neq \emptyset$, to uniquely defined values. The theorem immediately implies Theorem 1.2 of the introduction. The proof can be found in the full version of the paper. Once again, we illustrate our algorithm by applying it to *Up or Down?* of Figure 3 and for the third time derive the proper strategy for Player 1 in this game: We get $v_g = v_h = v_{h'} = 0$ and also $\underline{v}_g = \underline{v}_h = \underline{v}_{h'} = 0$ but $\overline{v}_h = 1$ and $\overline{v}_{h'} = 2$ and hence $\overline{v}_g = 1/(1+1/2) = 2/3$. Thus, the unique proper strategy for Player 1 is to choose U with probability $1/(1+1/2) = 2/3$ and D with probability $(1/2)/(1+1/2) = 1/3$.

Acknowledgements We would like to thank the three anonymous SODA reviewers for many detailed and useful comments and suggestions. Due to the space constraints, we were not able to make full use of all of these in this version of the paper, but they will be very useful for preparing the final version.

References

[1] Darse Billings, Neil Burch, Aaron Davidson, Robert Holte, Jonathan Schaeffer, Terence Schauenberg, and Duane Szafron. Approximating game-theoretic optimal strategies for full-scale poker. In *International Joint Conference on Artificial Intelligence*, pages 661-668, 2003.

[2] Lawrence Blume, Adam Brandenburger, Eddie Dekel. Lexicographic probabilities and equilibrium refinements. *Econometrica*, 59:81–98, 1991.

[3] Xi Chen and Xiaotie Deng. Settling the complexity of two-player Nash equilibrium. In *Proceedings of 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pages 261–272, 2006.

[4] Eric van Damme. A relation between perfect equilibria in extensive form games and proper equilibria in normal form games. *International Journal of Game Theory*, 13:1–13, 1984.

[5] Eric van Damme. *Stability and Perfection of Nash Equilibria*. Springer-Verlag, 2nd edition, 1991.

[6] Constantinos Daskalakis, Paul W. Goldberg, Christos H. Papadimitriou. The complexity of computing a Nash equilibrium. In *Procedings of the 38th Annual ACM Symposium on the Theory of Computing (STOC'06)*, pages 71–78, 2006.

[7] Melvin Dresher. *The Mathematics of Games of Strategy: Theory and Applications*. Prentice-Hall, 1961.

[8] Andrew Gilpin and Tuomas Sandholm. Lossless abstraction of imperfect information games. *Journal of the ACM*, 54(5), 2007.

[9] Elon Kohlberg and Jean-Francois Mertens. On the strategic stability of equilibria. *Econometrica*, 54:1003–1037, 1986.

[10] Daphne Koller, Nimrod Megiddo, and Bernhard von Stengel. Fast algorithms for finding randomized strategies in game trees. In *Proceedings of the 26th Annual ACM Symposium on the Theory of Computing (STOC'94)*, pages 750–759, 1994.

[11] Daphne Koller and Avi Pfeffer. Representations and solutions for game-theoretic problems. *Artificial Intelligence*, 94(1–2):167–215, 1997.

[12] David M. Kreps and Robert Wilson. Sequential equilibria. *Econometrica*, 50(4):863–894, July 1982.

[13] H.W. Kuhn. Extensive games and the problem of information. *Annals of Matematical Studies*, 28:193–216, 1953.

[14] Richard D. McKelvey, Andrew M. McLennan, and Theodore L. Turocy. Gambit: Software tools for game theory, version 0.97.0.7. http://econweb.tamu.edu/gambit, 2004.

[15] Peter Bro Miltersen and Troels Bjerre Sørensen. Computing proper equilibria of zero-sum games. In *Proceedings of the fifth International Conference on Computers and Games*, May 2006.

[16] Peter Bro Miltersen and Troels Bjerre Sørensen. Computing sequential equilibria for two-player games. In *Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'06)*, pages 107–116, 2006.

[17] Peter Bro Miltersen and Troels Bjerre Sørensen. Computing a quasi-perfect equilibrium of a two-player game. *Economic Theory*, to appear, 2008.

[18] R. B. Myerson. Refinements of the Nash equilibrium concept. *International Journal of Game Theory*, 15:133–154, 1978.

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

[20] Yoshitsugu Yamamoto. A path-following procedure to find a proper equilibrium of finite games. *International Journal of Game Theory*, 22:249–259, 1993.