Computing sequential equilibria for two-player games

(extended abstract)

Peter Bro Miltersen* and Troels Bjerre Sørensen*

Abstract

Koller, Megiddo and von Stengel showed how to efficiently compute minimax strategies for two-player extensive-form zero-sum games with imperfect information but perfect recall using linear programming and avoiding conversion to normal form. Koller and Pfeffer pointed out that the strategies obtained by the algorithm are not necessarily sequentially rational and that this deficiency is often problematic for the practical applications. We show how to remove this deficiency by modifying the linear programs constructed by Koller, Megiddo and von Stengel so that pairs of strategies forming a sequential equilibrium are computed. In particular, we show that a sequential equilibrium for a two-player zero-sum game with imperfect information but perfect recall can be found in polynomial time. In addition, the equilibrium we find is normal-form perfect. Our technique generalizes to general-sum games, yielding an algorithm for such games which is likely to be prove practical, even though it is not polynomial-time.

1 Introduction

1.1 Background and statement of main results

It has been known for more than fifty years [27, 36] 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 describing 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 [15] showed how to use *sequence form* representation to efficiently compute minimax 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 algorithm by Koller, Megiddo and von Stengel (henceforth, the KMvS-algorithm) has been used for constructing prescriptive strategies for concrete, often very large games, to be used in game playing software. It was been applied by Billings *et al* [2] to certain abstractions of heads-up Texas Hold'Em poker, each containing roughly ten million positions. Most recently, it was applied by Gilpin and Sandholm [8] to solve heads-up Rhode Island Hold'Em poker. The efficient algorithm based on sequence form representation and implemented using state of the art linear programming software was clearly essential for obtaining solutions for games this large.

Koller and Pfeffer [17] presented the software tool GALA, one of the applications of which was to encode and solve games using the KMvS-algorithm. Towards the end of their paper, they pointed out a certain deficiency of the strategies computed by the algorithm. Alex Selby [31], analyzing a simplified version of Texas Hold'Em poker using a variant of the algorithm (the strategy he computed was later used in the work by Billings *et al* mentioned above), found and described essentially the same deficiency. This deficiency and how to remedy it is the topic of the present paper. The deficiency may be summarized as follow: While the strategy computed by the KMvS-algorithm is a correct minimax strategy and thus guaranteed to attain an expected payoff of at least the game-theoretic value of the game considered, 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. In game theoretic terms, the computed strategy is not necessarily *sequentially rational*, a concept that has been thoroughly in-

*Department of computer science, University of Aarhus, Denmark. Email {bromille,trold}@daimi.au.dk. Supported by BRICS, a centre of the Danish National Research Foundation and by a grant from the Danish National Science Council.

vestigated; see the very comprehensive monograph by van Damme [35] for an overview. As Koller and Pfeffer correctly point out in their paper, there are several different (and to some degree competing) refinements of the concept of a Nash equilibrium meant to capture variations of this notion. However, a fairly permissive and hence relatively non-controversial refinement is the seminal notion of a *sequential equilibrium* due to Kreps and Wilson [18]. A sequential equilibrium is guaranteed to exist for any extensive form game. Also, it is an equilibrium in the usual sense, so for the case of zero-sum games, a sequential equilibrium will still be a pair of maximin/minimax strategies. Most importantly, the intention of the notion is that in a sequential equilibrium, at every situation in the game, mistakes made by the opponent in the past are exploited optimally, relative to some consistent ("sensible") *belief* about the situation. Thus, intuitively, a strategy prescribed by a sequential equilibrium cannot "return gifts".

Insisting on sequentiality removes many intuitively insensible equilibria but not all: The notion is only concerned with exploiting mistakes the opponent made in the past and does not try to deal with mistakes he may make in the future. An additional "niceness" property which rules some additional insensible behavior not ruled out by sequentiality is Selten's notion of *normal-form perfection* [32, 9]. For a two-player game, an equilibrium is normal-from perfect if and only if none of the two strategies it prescribes are dominated (van Damme [35, Theorem 3.2.2]). The notions of sequentiality and normal-form perfection are incomparable, i.e., neither is implied by the other.

For illustration, we present two simple zero-sum games with equilibria exhibiting anomalies similar to those described by Koller and Pfeffer and by Selby, and explain on an intuitive level how insisting on sequentiality and normal-form perfection of the equilibria computed would resolve the anomalies. First, consider the following game, *Guess-the-Ace*, played by Player I and Player II. A standard deck of 52 cards is shuffled perfectly by a dealer. Player I may now choose to end the game, in which case no money is exchanged between the two players. Player I's other option is to offer Player II $1000 for correctly answering the following question: When the dealer reveals the top card of the deck, will it be the ace of spades? If asked, Player II may answer either "yes" or "no" and the game ends by the dealer revealing the top card and Player I paying Player II if he guessed correctly. Intuitively, it seems obvious that in any sensible strategy for Player II, he should guess (if asked) that the top card of the deck is *not* the ace of spades. After all, this is the case with probability 51/52. Indeed, the unique sequential equilibrium for Guess-the-Ace prescribes that Player II should guess that the top card is not the ace of spades, with probability 1.

The extensive form for this game, drawn by using game theory software tool Gambit [22] is given in Figure 1. The probabilities being written as labels on the arcs of the game is the Nash Equilibrium found by Gambit using its implementation of the KMvS-algorithm. Note that in the equilibrium found, Player II guesses with probability 1 that the top card *is* the ace of spades! Though describing a clearly insensible strategy for Player II, the strategy profile computed *is* a Nash equilibrium as it prescribes for Player I to (sensibly) stop the game without asking the question and risking $1000. *Any* strategy for Player II is in equilibrium with this strategy. The equilibrium found is subgame perfect and is found even if Gambit is asked to solve subgames separately (as the game has no proper subgames). But imposing the constraint of sequentiality would rule out this insensible equilibrium. In fact, imposing the constraint of normal-form perfection would as well: The strategy of Player II of guessing that the top card *is* the ace of space is dominated by guessing that it is *not*. We may modify the example into one where sequentiality eliminates the insensible behavior but normal-form perfection does not, by adding a second move of Player I *after* the guess of Player II: We give him at this point in time the option of giving Player II another gift of $1000, no strings attached this time. Now normal-form perfection does not rule out Player II guessing that the top card is the ace of space, but sequentiality still does: In a normal-form perfect equilibrium Player II is allowed to hope that the second gift will be given if and only if he makes the insensible guess, in a sequential equilibrium he is not.

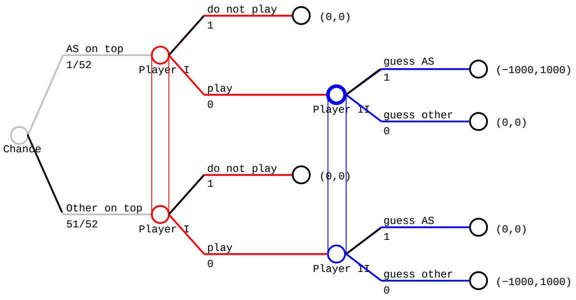

Figure 1: Non-sequential equilibrium for "Guess-the-Ace" found by the KMvS-algorithm.

To give an example where normal-form perfection rules out a particular piece of insensible behavior but sequentiality does not, we consider a variant of the celebrated example of three-card poker due to Kuhn [19]. See e.g. Chvátal [4, pages 235–247] for a textbook account. In Kuhn's original version, Player I and Player II both pay an ante of $1 and are then each dealt a card from a three-deck card containing an ace, a king and a queen. Player I must now either *bet* $1 that his card is the highest or *check* to Player II. If he bets, Player II may choose to *call* the bet or *fold*. If Player I checks, player II can either check himself, or bet. If he bets, player II can either call the bet or fold. If a player folds, the other player wins the pot. If no player folds, the cards are revealed and the player with the highest card wins the pot.

The variation we consider consists of adding a third option for the players: In addition to calling a bet or folding to a bet, one is allowed to *raise* the bet by an additional $1, which the other player now must call or fold to. Such an option is of course standard in real versions of poker. We only allow a single raise as this extension is enough to illustrate our point. Thus, the final size of the pot can be at most $6, including the antes. The extensive form of the resulting game is too big to be presented graphically here (it contains 91 positions, including terminal ones). In Figure 2, we show a single branch of the game, omitting information sets, namely the one where Player I is dealt the ace and Player II is dealt the queen. The probabilities on the arcs are from the (reduced) behavioral strategies computed using Gambit's implementation of the KMvS-algorithm. We see that Player I starts by checking his ace with probability 1. This behavior, though perhaps slightly surprising, is in fact perfectly sensible (Player I does this on all hands, taking some of the positional advantage away from Player II). The insensible behavior occurs if Player II then bets: Player I then calls instead of raising. This is clearly insensible as Player I knows for sure that he has the best hand. Still, the strategy profile computed is a Nash equilibrium: Player I knows that a sensible opponent would never call the raise, so from a min-max perspective there is no reason to make it. Unlike in the example of Guess-the-Ace, here the insensible behaviour is made in an information set which will be reached with strictly positive probability if both players play their equilibrium strategies, as Player II bets his queen with probability 1/3 when checked to, as a bluff. And indeed, the strategies computed *do* form a reduced sequential equilibrium, but it is not normal-form perfect: A normal-form perfect equilibrium must raise the ace with probability 1, as raising dominates calling.

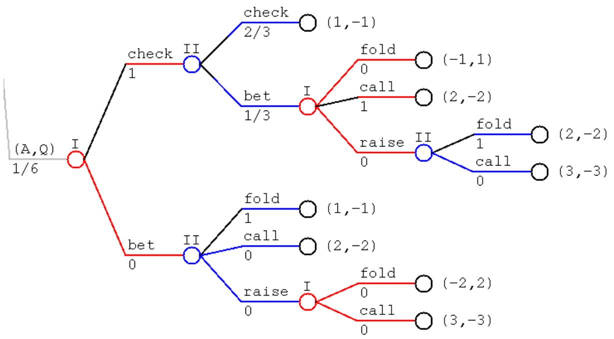

Figure 2: Is the pot really big enough!?

Our discussion motivates the main theoretical result of the present paper, strengthening the result of Koller, Megiddo and von Stengel.

Theorem 1.1. *A sequential and (simultaneously) normal-form perfect equilibrium for a given extensive-form, two-player, zero-sum game with imperfect information but perfect recall may be computed in polynomial time.*

Our strategy for proving this is to modify the linear programs of Koller, Megiddo and von Stengel and solve the resulting modified programs. The modified programs can be viewed as (symbolically or actually) *perturbed* versions of the original programs. This approach not only leads to a polynomial time algorithm, but also to a (different) algorithm that is quite efficient in practice. Indeed, we have made a provisional implementation of our algorithm and preliminary computational experiments are quite encouraging, efficiency-wise. As stated above, implementations of the KMvS-algorithm can be and have been practically applied to very large games and we think the algorithm of this paper *can* be similarly applied to large games. As the deficiency pointed out by Koller and Pfeffer and by Selby is a very real issue that occurs in practice when the KMvS-algorithm is applied, we also think that a variant ensuring sequential rationality such as ours *should* be applied in such settings, in order to ensure that strategies prescribing sensible play are computed!

While our main focus in this extended abstract is on the zero-sum case, motivated by the applications in artificial intelligence, the practical variant of our algorithm generalizes to general-sum games. We describe this in the full version of the paper, a preliminary version of which can be found on the homepage of the first author. Again, we follow the approach of Koller, Megiddo and von Stengel [15, 16], but now perturb their linear complementarity programs instead of their linear programs and solve the perturbed games using Lemke's algorithm. Thus, the full version of this paper also describes a practical (but not polynomial-time) algorithm for finding a sequential and normal-form perfect equilibrium of a two-player general-sum game with imperfect information but perfect recall.

1.2 Previous research using related techniques As stated above, our technique is based heavily on the sequence form technique of Koller, Megiddo and von Stengel. However, as they state it, their technique only provides *reduced* behavioral strategies, and to produce a sequential equilibrium, one has to specify meaningful behavior at every information set, also those off the equilibrium path. Discussing this issue, McKelvey and McLennan [20, section 4.3] hypothesized "that a suitable generalization of the sequence form will be the natural vehicle for computation of sequential equilibrium." It is exactly this hypothesis that we confirm in this paper. Namely, the versions of our algorithm described in Section 2.3 for the zero-sum case and in the full version of the paper for the general-sum case use *lexicographic perturbations* of the linear programs and linear complementarity programs associated with the sequence form, i.e., perturbations which are formal polynomials in an indeterminate ϵ. We can informally think of ϵ as a value which is bigger than 0 but infinitesimally small. Such a perturbation is called lexicographic as two perturbed values may be compared by comparing the two vectors of coefficients of the polynomials using the lexicographic ordering. The lexicographic perturbation technique was originally introduced as a technical device for eliminating degeneracies in tableaus occurring during execution of the simplex algorithm for solving linear programs in the early 1950s [3, 6]. It was later applied in computational game theory for similarly eliminating degenerate situations when solving linear complementarity programs using the Lemke-Howson algorithm and Lemke's algorithm [7, 16, 40, 20, 39, 23]. The lexicographic perturbation method was also used by Wilson [41] to find *simply stable* equilibria of normal-form games. In contrast to the applications mentioned above, one may characterize Wilson's lexicographic perturbations as being "meaningful" or "semantic": In addition to their use for degeneracy elimination, Wilson's perturbations have a game-theoretic interpretation in terms of perturbed *games*. Perturbed games being ubiquitous in the theory of equilibrium refinements, it does indeed seem very natural to explore the use of symbolic perturbations when computing such refined equilibria and this approach is further explored in this paper. Our use of lexicographic perturbations is *purely* semantic, i.e., the perturbations are not meant to care of degeneracies at all (these we handle separately) but should be interpreted in terms of a perturbed game where only behavioral strategies that are fully mixed at every information set are permitted.

1.3 Previous research solving related problems von Stengel, van den Elzen and Talman [40] showed how to apply Lemke's algorithm to find a normal-form perfect equilibrium for an extensive-form general-sum game. We similarly apply Lemke's algorithm to find an equilibrium which is normal-form perfect as well as sequential (in the full version of the paper) and thus give an alternative to their algorithm. It is interesting that technically our algorithm is much closer related to the original application of Lemke's algorithm by Koller, Megiddo and von Stengel than to the application of Lemke's algorithm by von Stengel, van den Elzen and Talman.

Some theoretical work has been done on computing sequential equilibria for general-sum games [1, 13]. The algorithms presented in these works are worst-case exponential time, as no known procedure guaranteed to find even a Nash Equilibrium in a general-sum game (even in normal form) is known to be polynomial time [10, 28, 5, 30]. In addition to these theoretical results, the game theory software tool Gambit [22] has a procedure for computing a sequential equilibrium in general-sum games. The procedure is oblivious as to whether the input is zero-sum or not. It is computing the *logit solution* of the game, a notion due to McKelvey and Palfrey [21]. This is, interestingly, also a notion of equilibria of perturbed games, but different from our perturbed games. To compute the logit solution, the *payoffs* are (randomly) perturbed, separately for each of the two players and hidden for the other player. This perturbation thus turns a zero-sum game into a non-zero sum game. The associated procedure is using floating point arithmetic and numerical instability causes it sometimes to fail computing a sequential equilibrium [33]. Even for quite small games, the procedure runs very slowly and it seems out of the question that it could be used to solve games the size of the games considered by, say, Billings *et al.* In the three card poker example presented above, Gambit (a library developed over more than a decade and coded in C++) used 75 seconds to find a sequential equilibrium using floating point arithmetic, while the preliminary implementation of our algorithm for the zero-sum case (coded by the present authors in java) used 6 seconds (including the time required to load java libraries, etc.) to find a sequential equilibrium using exact rational arithmetic. We have not yet implemented our algorithm for the general-sum case, but as it is based on Lemke's algorithm which has a reputation of being practical [40], we suspect that our method will be practical, also in the general-sum case.

1.4 Organization of the remaining paper Section 2 is the bulk of the paper, describing our algorithm for the zero-sum case. In Section 2.1, we outline the main ideas of the KMvS-algorithm while reminding the reader about the basic concepts concerning extensive-form games. In Section 2.2, we state the Kreps-Wilson definition of a sequential equilibrium. In Section 2.3, we show how to modify the linear programs of Koller, Megiddo and von Stengel and use the modified programs to identify a sequential equilibrium for the given game and show how to compute it efficiently, in a theoretical and a practical sense. Due to space constraints, almost all proofs are omitted. They may be found in the full version of the paper, a preliminary version of which is available at the first author's homepage. The full version of the paper also describes our algorithm for the general-sum case.

2 Finding a sequential equilibrium using linear programming

2.1 Extensive-form games and the sequence form By a *game G* we mean in the following a two-player extensive-form zero-sum game with imperfect information but perfect recall. We will name the two players Max and Min, with Max attempting to maximize the payoff and Min attempting to minimize it. The game is given as a game tree with information sets in the standard way (see, e.g., [15] or any textbook on game theory).

In order to formally argue that the equilibria we consider can be found in polynomial time, we assume that all payoffs of G and all probabilities associated with chance nodes are rational numbers, given as fractions. Also, we fix some encoding $\mathrm{enc}(G)$ of G as a string over a finite alphabet using some standard way of representing trees and information sets and representing rational numbers as fractions. We assume that the length $|\mathrm{enc}(G)|$ of the encoding is at least 2 for any game. Finally, we say that a rational number p/q occurring in our derivation is *simple*, if the absolute values $|p|, |q|$ are less than $2^{|\mathrm{enc}(G)|^c}$ for some constant c, independent of G, i.e., if the description of the number as a fraction has size polynomial in the length of the encoding of G.

Given a game G, a *behavioral strategy* for Max (Min) in G is a family of probability distributions, one for each node belonging to Max (Min), on the outgoing arcs (representing actions) from that node. For any two nodes belonging to the same information set, the two corresponding probability distributions must be identical. A *reduced* behavioral strategy is defined similarly, except that a node z belonging to Max (Min) is not assigned a probability distribution in a reduced strategy if some action belonging to Max (Min) on the path from the root to z has been assigned probability 0. A *strategy profile* is a behavioral strategy for each of the two players.

An important insight of Koller, Megiddo and von Stengel [14, 37, 38, 15] and earlier and independently Romanovskii [29] is that behavioral strategies are for computational purposes often better represented in *sequence form* which we describe next. Given a behavioral strategy for one of the players, the corresponding vector of *realization weights* is the following assignment of real numbers to each node z in G: We assign to z the product of behavioral probabilities of actions belonging to the player and appearing on the path from the root of the game to z. For games of perfect recall, all nodes in an information set are assigned the same realization weight. Clearly, the map from reduced behavioral strategies to vectors of realization weights is 1-1, so given a vector of realization weights (a *realization plan*), we may talk about the corresponding reduced behavioral strategy. We shall later use the fact that each behavioral probability is the ratio between two realization weights.

The crucial observations of Koller, Megiddo and von Stengel are the following:

1. *For a game of perfect recall, the statement that a vector is a behavioral plan corresponding to some behavioral strategy can be expressed by a (short) system of linear equations, containing coefficients from* $0, 1, -1$ *only.* In the following, for a fixed game G, we let $Ex = e$ be the equations expressing that x is a realization plan for Max and we let $Fy = f$ be the equations expressing that y is a realization plan for Min.

2. *If Max plays a strategy with realization plan x and Min plays a strategy with realization plan y, then the expected payoff for Max is given by a bilinear form $x^{\top}Ay$.* Here, A is a matrix, depending on G, containing simple rationals.

These observations are the basis for the derivation of the linear programs of Koller, Megiddo and von Stengel expressing a pair of minimax and maximin strategies for the two players. Due to lack of space, we do not describe the derivation. However, in Section 2.3, we make an analogous derivation for a certain perturbed game $G(\epsilon)$ which is identical to the derivation of Koller, Megiddo and von Stengel for $\epsilon = 0$. Since linear programs are polynomial time solvable [12, 11], there is a polynomial time algorithm for finding the equilibrium the programs describe. This is the equilibrium returned by the KMvS-algorithm.

2.2 The notion of a sequential equilibrium The definition of a sequential equilibrium due to Kreps and Wilson [18] is based on the notion of *beliefs*. Formally, a belief of a player is a probability distribution on each of his information sets. Intuitively, the belief should indicate the subjective probability of the player of being in each of the nodes in the information set, given that he has arrived at this information set. An *assessment* (ρ, μ) is a strategy profile ρ, and a *belief profile* μ: a belief for each of the two players. A sequential equilibrium is an assessment which is (1) *consistent* and (2) *a sequential best reply against itself*, the former notion capturing that the beliefs are sensible given the strategies, and the latter notion capturing that the strategies are sensible given the beliefs. We define these two notions formally next, for the case of zero-sum games with perfect recall.

We first define consistency for *fully mixed* strategy profiles, i.e., ones where every action in every information set has a strictly positive probability of being taken. For such a strategy profile, the *induced belief profile* is the unique one consistent with the strategy profile: The two strategies being played out against each other induces a probability distribution on possible plays; the induced belief assigns to information set u the conditional probability distribution on u derived from this probability distribution. This is well-defined as at most one node in u may be reached during each particular play (due to the perfect recall property) and u has a nonzero probability of being reached (as the strategies are fully mixed). The crucial insight of Kreps and Wilson is a generalization of this consistency notion to strategy profiles where some of the information sets may be reached with probability 0: For this general case, we say that an assessment is consistent if it is the limit point (in Euclidean distance) of a sequence of consistent assessments with fully mixed strategy profiles, i.e., a limit point of a sequence $(\rho_n, \mu_n), n = 1, 2, \ldots$, so that ρ_n is a completely mixed strategy profile and μ_n is the induced belief profile.

We next define what it means to be a sequential best reply against itself. First we define the notion of the *value* $\text{val}^{\rho}(z)$ of a node or leaf z in the game tree, given a strategy profile ρ. Informally, the value of z is the expected payoff for a play starting in that node and played according to ρ. Formally, the value of each leaf z is defined to be the payoff associated with z. The value of each node z is defined recursively by

$$\text{val}^{\rho}(z) = \sum_{j} \rho(j)\text{val}^{\rho}(s_j(z)) \tag{2.1}$$

where the sum is over the possible actions j at z, $\rho(j)$ is the behavioral probability of action j in the profile ρ (if the node belongs to nature, we define it to be nature's probability that this action is taken) and $s_j(z)$ is the j'th successor of z, i.e., the node or leaf in the game tree that will be reached if action j is taken at node z. An assessment (ρ, μ) is a sequential best reply against itself if

1. for every information set u belonging to Max and every action j at u for which $\rho(j) > 0$, we have that $\sum_{z\in u} \mu(z)\text{val}^{\rho}(s_j(z))$ is at least as big as $\sum_{z\in u} \mu(z)\text{val}^{\rho}(s_i(z))$ for any other action i at u and

2. for every information set u belonging to Min, and every action j at u for which $\rho(j) > 0$, we have that $\sum_{z\in u} \mu(z)\text{val}^{\rho}(s_j(z))$ is no bigger than $\sum_{z\in u} \mu(z)\text{val}^{\rho}(s_i(z))$ for any other action i at u.

2.3 Identifying a sequential equilibrium The definition of a sequential equilibrium for a game G suggests looking at limit points of a sequence of assessments which are not necessarily equilibria for G. This is indeed our strategy. Given a game G, the sequence we shall look at shall be equilibria for a certain perturbed game, $G(\epsilon)$, for a parameter $\epsilon > 0$ and we shall consider limit points as $\epsilon \to 0$.

We obtain the perturbed game $G(\epsilon)$ from G by restricting the set of valid realization plans for both players. Specifically, if u is an information set belonging to player j, *we demand that the realization weight of u is at least ϵ^{d_u}, where d_u is the number of actions performed by player j before arriving in information set u.* Choosing this exact perturbation will ensure that a limit point of basic (in the sense of linear programming) equilibria of the perturbed game will be a sequential best reply against itself *and* efficiently computable, as will be seen below.

Let k_ϵ be the vector indexed by information sets u of Max, so that $(k_\epsilon)_u = \epsilon^{d_u}$, where d_u is the number of actions performed by Max before arriving in information set u. Suppose that a strategy of Min is fixed and given in sequence form by a realization plan y. A best response by Max as a realization plan x is then given by

$$\begin{array}{cccc} & \max_x & x^\top(Ay) & \text{so that} \\ (2.2) & Ex & = & e \\ & x & \geq & k_\epsilon \end{array}$$

Note that the behavioral strategy corresponding to a solution of (2.2) has to be fully mixed at every information set. The dual of (2.2) is

$$\begin{array}{cccc} & \min_{p,u} & p^\top e - u^\top k_\epsilon & \text{so that} \\ (2.3) & E^\top p & \geq & Ay + u \\ & u & \geq & 0 \end{array}$$

The program (2.3) expresses the expected payoff that Max can achieve in the perturbed game $G(\epsilon)$ if Min plays using strategy y. Min wants to choose a valid realization plan y so that this is minimized. His minimax strategy is given by

$$\begin{array}{lccl} & \min\limits_{p,u,y} & p^\top e - u^\top k_\epsilon & \text{so that} \\ (2.4) & E^\top p & \geq & Ay + u \\ & Fy & = & f \\ & y & \geq & l_\epsilon \\ & u & \geq & 0 \end{array}$$

where l_ϵ is defined analogously to k_ϵ, replacing Max with Min. Note that it is obvious that (2.4) has a feasible solution for all sufficiently small $\epsilon > 0$.

Reversing the roles of the two players and arguing completely analogously, we obtain the maximin strategy for Max.

$$\begin{array}{lccl} & \max\limits_{q,v,x} & q^\top f + v^\top l_\epsilon & \text{so that} \\ (2.5) & F^\top q & \leq & A^\top x - v \\ & Ex & = & e \\ & x & \geq & k_\epsilon \\ & v & \geq & 0 \end{array}$$

Observing that (2.4) and (2.5) are LP duals we have found an equilibrium for the perturbed game $G(\epsilon)$.

For expositional reasons, we now make an assumption about the game which can be made without loss of generality: By adding a constant to the payoff of every leaf of the original extensive form, we may transform the game into one where every payoff for Max is positive and every payoff for Min is negative, and we will assume that this is indeed the case. Under this assumption, a reformulation of (2.4) is

$$\begin{array}{lrcl} & \max\limits_{p,u,y} & -p^\top e + u^\top k_\epsilon & \text{so that} \\ (2.6) & -E^\top p + Ay + u & \leq & 0 \\ & -Fy & \leq & -f \\ & -y & \leq & -l_\epsilon \\ & p, u, y & \geq & 0 \end{array}$$

i.e., a standard form linear program in the terminology of Chvátal [4], which we shall use henceforth. We can make a similar reformulation of (2.5). By introducing slack variables, we put (2.6) in the following form.

$$\begin{array}{lccl} & P_\epsilon : \quad \max\limits_{x'} & {c_\epsilon}^\top x' & \text{so that} \\ (2.7) & A'x' & = & b_\epsilon \\ & x' & \geq & 0 \end{array}$$

Here, b_ϵ and c_ϵ are vectors with entries being either constants (i.e., not depending on ϵ) or powers ϵ^j of ϵ where j is at most d, the maximum possible number of actions in any play of the game. The matrix A' has entries which are either entries from A or in $\{-1, 0, 1\}$.

We may put the reformulation of (2.5) in the same format to obtain Q_ϵ. For every statement we make below about P_ϵ, the corresponding statement holds for Q_ϵ, by syntactic similarity.

Suppose we have a basic (but not necessarily feasible) solution x^* to P_ϵ for any particular $\epsilon > 0$ and let another value $\delta > 0$ be given. We define the *corresponding* basic (but not necessarily feasible) solution to P_δ to be the basic solution with the same set of variables in the basis as x^*.

LEMMA 2.1. *For any game G, there is a simple rational number ϵ_G so that for any ϵ, δ with $0 < \epsilon, \delta < \epsilon_G$ the following holds:*

1. *P_ϵ has a feasible solution.*
2. *Let x' be a basic (not necessarily feasible) solution to P_ϵ. Then x' is feasible and optimal if and only if the corresponding solution to P_δ is feasible and optimal.*

Also, the corresponding statements holds with P_ϵ replaced with Q_ϵ.

Proof. Any basic solution $x' = \binom{x_B}{x_N}$ to P_ϵ where x_B is the basic part and $x_N = 0$ is the non-basic part, is by definition (e.g., Chvátal [4, page 100]) given by a tableau

$$x'_B = B^{-1}b_\epsilon - B^{-1}Dx_N \tag{2.8}$$

where D and B are submatrices of A' and the value of the objective function is given by

$$z = c_B B^{-1} b_\epsilon + (c_N - c_B B^{-1} D)x_N, \tag{2.9}$$

with $c_\epsilon = \binom{c_B}{c_N}$. The entries in A' and hence D are simple rational numbers. Cramer's rule implies that the entries in B^{-1} are also simple rational numbers, B being a submatrix of A'.

The solution x' is feasible if and only if all entries in $B^{-1}b_\epsilon$ are greater than or equal to 0 and it is optimal if and only if all entries in $c_N - c_B B^{-1} D$ are smaller than or equal to 0. By the above bounds on the entries of the matrices, this condition is equivalent to a system of linear inequalities

$$\forall i : \sum_{j=0}^{d} k_{ij}\epsilon^j \geq 0 \tag{2.10}$$

where k_{ij} are integers of absolute value less than some simple integer K. We let $\epsilon_G = 1/(2K)$. Now, for any $\epsilon < \epsilon_G$, the inequality (2.10) is equivalent to

$$(2.11) \qquad \forall i : [\forall j : k_{ij} = 0] \vee [k_{\min\{j|k_{ij} \neq 0\}} > 0]$$

which does not depend on ϵ, as was to be proved.

We are now ready to identify the equilibrium we compute in this paper. Given a game G, let $\epsilon < \epsilon_G$. Let x^* be a feasible and optimal basic solution to P_ϵ and let y^* be a feasible and optimal basic solution to Q_ϵ. For any $\delta < \epsilon_G$, let x^*_δ be the solution to P_δ corresponding to x^* and let y^*_δ be the solution to Q_δ corresponding to y^*. Also, let π^*_δ be the reduced behavioral strategy corresponding to the realization plan found in x^*_δ and let σ^*_δ be the reduced behavioral strategy corresponding to the realization plan found in y^*_δ. Let $\rho_\delta = (\pi^*_\delta, \sigma^*_\delta)$ and let μ_δ be the induced belief profile corresponding to ρ_δ. Note that μ_δ is well-defined as π^*_δ and σ^*_δ are fully mixed.

Lemma 2.2. *$(\rho_\delta, \mu_\delta)$ converges as $\delta \to 0+$ in Euclidean distance to an assessment (ρ, μ).*

Proof. To establish convergence, we use that each entry of π^*_δ (resp., σ^*_δ) is a ratio between two realization weights which are entries in x^*_δ (resp., y^*_δ). But by equation (2.8), each such entry is given by a polynomial in δ. Thus, each entry of π^*_δ and σ^*_δ are *rational* functions of δ and since we furthermore have that they are values in the interval $(0, 1)$ (as they are behavioral probabilities in the perturbed game $G(\delta)$) each must converge to a value in $[0, 1]$ as $\delta \to 0$. As each probability in the induced beliefs are products of the behavioral probabilities and probabilities of actions of nature, a similar argument applies to show that μ_δ converges to a belief profile μ.

In the full version of the paper, we prove that the assessment (ρ, μ) of Lemma 2.2 is a sequential and normal-form perfect equilibrium. It fact, it can be shown that the equilibrium is *quasi-perfect*, a common generalization of these two concepts due to van Damme [34]. Also, note that unlike the equilibrium defined by the original linear programs by Koller, Megiddo and von Stengel, the equilibrium just described prescribes behavior in *all* information sets, including those off the equilibrium path. We next describe algorithms for computing it.

The only merit of the first algorithm we describe is that it is polynomial time. We simply run a polynomial time LP solver on the program P_ϵ with the parameter ϵ being fixed to the value $\epsilon_G/2$ established in Lemma 2.1. We can assume that the algorithm outputs a combinatorial description of a basic optimal solution (i.e., a partition of the variables into basic and non-basic ones), which enables us to find a symbolic description of the corresponding solution to P_δ as a polynomial in δ by computing the inverse of the basis B^{-1} and using equation (2.8). We then compute symbolic expressions for the corresponding behavioral strategies and beliefs (each value being a rational function in δ) and finally find the sequential equilibrium by setting $\delta = 0$ in these symbolic expressions. This is a very impractical algorithm, since ϵ_G is very small (even if we had bothered to compute and state the best possible bound on ϵ_G, which we haven't, for precisely this reason), meaning that one would have to run the linear program on an input with very large coefficients (containing a number of digits bigger than the size of G). Nevertheless, as ϵ_G expressed as a fraction does have polynomial length, it is a polynomial algorithm, thus establishing Theorem 1.1.

We next discuss a much more practical approach. The equilibrium described above can be found by establishing a feasible and optimal basic solution to (P_ϵ, Q_ϵ) as a function of $\epsilon > 0$ and then considering the limit as $\epsilon \to 0$. A practical way of doing this is by a symbolic execution of the simplex algorithm with the parameter of perturbation ϵ being kept as an indeterminate. As is apparent from equation (2.8), each tableau in such an execution will have entries in the leftmost column and in the row of the cost function being formal polynomials in ϵ of degree at most the depth of the extensive form, with the rest of the tableau containing standard rational values. We can decide if we have a terminal tableau and whether or not a given pivot is allowed by checking if a system of equations such as (2.10) is true which, for sufficiently small ϵ, is equivalent to a *lexicographic rule* such as (2.11), which is easily checked symbolically. When a terminal tableau is reached, we may find each behavioral probability of the limit point corresponding to $\epsilon \to 0$ by inspecting the most significant non-zero terms (with higher degree terms having lower significance) of the two polynomials representing the relevant realization weights. The belief of the limit point can be found similarly. Note that the algorithm is extremely similar to the lexicographic perturbation method for ensuring termination of the simplex algorithm in degenerate situations (see, e.g., Chvátal [4, pages 34–37]), the most striking difference being that we here have a *dual* as well as a primal symbolic perturbation, i.e., symbolic entries in a row of the tableau as well as in a column rather than in a column only. Here, however, we are using it for a completely different purpose and as we have stated it, it does *not* take care of degeneracies for us, i.e., the symbolically perturbed tableaus may still very well be degenerate. If such degenerate tableaus occur, we can apply a pivoting rule such as Bland's to guarantee termination of the algorithm. As the simplex algorithm is worst case exponential, the algorithm is not polynomial time. On the other hand, it is quite practical and easy to implement. As a proof-of-concept, we have made a preliminary implementation in java (using exact rational arithmetic). Preliminary experiments seem promising efficiency-wise (but are not part of this paper). The code and demo programs finding sequential equilibria in the games *Guess-the-Ace* and *Three card poker with raise* explained in the introduction are available at `http://www.daimi.au.dk/~trold/gtf/`

3 Conclusions and open problems

We have established that a sequential and normal-form perfect equilibrium of a two-player game with perfect recall can be found efficiently in a practical sense for the zero-sum case as well as for the general-sum case (in the full version of the paper) and also efficiently in a theoretical sense, for the zero-sum case. In fact, the equilibrium we compute is *quasi-perfect*, a common generalization of the notions of sequentiality and normal-form perfection due to van Damme [34].

An alternative equilibrium refinement notion is *extensive-form perfection* (see, e.g., van Damme, [35, pages 113ff]). It was shown by Mertens [24] that insisting on an extensive-form perfect equilibrium is, in general, inconsistent with insisting on a normal-form perfect one: He exhibits a game where no equilibrium has both properties. His example is not a zero-sum game, so one may still ask if our algorithm for the zero-sum case always computes an extensive-form perfect equilibrium. This is *not* the case, as can be seen from the "solitaire" example in Figure 3. In this example, Player I is the only one who gets to move and he can choose between a safe action that immediately leads to a payoff equal to the value of the game and a risky action which will lead to the same payoff, if Player I avoids making a mistake in his next move. In the unique extensive-form perfect equilibrium for the game, the safe action is taken with probability 1; intuitively, a player playing a strategy in an extensive-form perfect equilibrium must not only try to set the opponent up to make a mistake, but must also worry about the possibility that he *himself* makes a mistake later on. However, as can be easily checked, the strategy taking the risky action with probability 1 is a limit point of basic optimal solutions to our perturbed linear programs for this game and hence this is a possible output of our algorithm. As a theoretical open problem we thus leave: *Can an extensive-form perfect equilibrium of a two-player zero-sum game with perfect recall be found in time polynomial in the size of the given extensive form?*

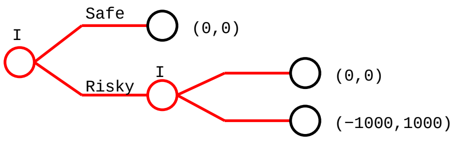

Figure 3: Safe-or-risky

An even more intriguing question is: *Can a normal-form proper equilibrium* (in the sense of Myerson [26], see also van Damme [35, pages 57ff]) *of a two-player zero-sum game with perfect recall be found in time polynomial in the size of the given extensive form?* This latter question is particularly interesting for the applications in artificial intelligence: Insisting on a proper equilibrium would rule out some arguably insensible behavior which is not ruled out by insisting on quasi-perfection. For further discussion of this, see our recent paper [25].

Acknowledgements We would like to thank Daphne Koller, Bernhard von Stengel, Andrew McLennan, Robert Wilson, Ted Turocy and an anonymous SODA reviewer for very helpful discussions and comments on previous versions of this manuscript.

References

[1] S. Azhar, A. McLennan, and J.H. Reif. Computation of equilibria in noncooperative games. Technical Report CS-1991-36, Duke University, 1991. Proc. Workshop for Computable Economics, Dec. 1992. To appear in Computers & Mathematics with Applications, 2005.

[2] 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 *Proceedings of the 18th International Joint Conference on Aritificial Intelligence (IJCAI-03)*, 2003.

[3] A. Charnes. Optimality and degeneracy in linear programming. *Econometrica*, 20:160–170, 1952.

[4] V. Chvátal. *Linear Programming.* W. H. Freeman, 1983.

[5] Vincent Conitzer and Tuomas Sandholm. Complexity results about Nash equilibria. In *Proceedings of the 18th International Joint Conference on Aritificial Intelligence (IJCAI-03)*, pages 765–771, Acalpulco, Mexico, 2003.

[6] G.B. Dantzig, A. Orden, and P. Wolfe. The generalized simplex method for minimizing a linear form under linear inequality restraints. *Pacific Journal of Mathematics*, 5:183–195, 1955.

[7] B. Curtis Eaves. The linear complementarity problem. *Management Science*, 17(9):612–634, 1971.

[8] Andrew Gilpin and Tuomas Sandholm. Finding equilibria in large sequential games of imperfect information. Technical Report CMU-CS-05-158, Carnegie Mellon University, Pittsburgh, PA, 2005.

[9] J. Harsanyi and R. Selten. *A General Theory of Equilibrium Selection in Games.* MIT Press, 1988.

[10] David S. Johnson, Christos H. Papadimitriou, and Mihalis Yannakakis. How easy is local search? *Journal of Computer and System Sciences*, 37(1):79–100, August 1988.

[11] N. Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4(4):373–395, 1984.

[12] L.G. Khachiyan. A polynomial algorithm in linear programming. *Soviet Mathematics Doklady*, 20:191–194, 1979.

[13] Elon Kohlberg and Philip J. Reny. Independence on relative probability spaces and consistent assessments in game trees. *Journal of Economic Theory*, 75(2):280–313, 1997.

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

[15] 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*, pages 750–759, 1994.

[16] Daphne Koller, Nimrod Megiddo, and Bernhard von Stengel. Efficient computation of equilibria for extensive form games. *Games and Economic Behavior*, 14:247–259, 1996.

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

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

[19] H.W. Kuhn. A simplified two-person poker. In H. W. Kuhn and A. W. Tucker, editors, *Contributions to the theory of games I*, volume 24 of *Annals of Mathematical Studies*. 1950.

[20] Richard McKelvey and Andrew McLennan. Computation of equilibria in finite games. In J. Rust H. Amman, D. Kendrick, editor, *Handbook of Computational Economics*, pages 87–142. Elsevier, 1996.

[21] Richard McKelvey and Tom Palfrey. Quantal response equilibria for extensive form games. *Experimental Economics*, 1:9–41, 1998.

[22] 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.

[23] Andrew McLennan and Rabee Tourky. From imitation games to Kakutani. Manuscript, 2005.

[24] Jean-Francois Mertens. Two examples of strategic equilibrium. *Games and Economic Behavior*, 8:378–388, 1995.

[25] Peter Bro Miltersen and Troels Bjerre Sørensen. Finding proper equilibria in matrix games efficiently. Manuscript, available at http://www.daimi.au.dk/˜bromille/Papers/, 2005.

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

[27] J. F. Nash. Equilibrium points in n-person games. *Proc. Nat. Acad. Sci. U.S.A.*, 36:48–49, 1950.

[28] C. H. Papadimitriou. On graph-theoretic lemmata and complexity classes. In *Proceedings of the 31st Annual Symposium on Foundations of Computer Science*, pages 794–801, St. Louis, MS, October 1990. IEEE Computer Society Press.

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

[30] R. Savani and B. von Stengel. Exponentially many steps for finding a Nash equilibrium in a bimatrix game. In *Proceedings to 45th Annual IEEE Symposium on Foundations of Computer Science (FOCS04)*, pages 258–267, 2004.

[31] Alex Selby. Optimal heads-up preflop holdem. Webpage, http://www.archduke.demon.co.uk/ simplex/index.html.

[32] Reinhard Selten. A reexamination of the perfectness concept for equilibrium points in extensive games. *International Journal of Game Theory*, 4:25–55, 1975.

[33] Theodore L. Turocy. Personal communication.

[34] 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.

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

[36] J. von Neumann and O. Morgenstern. *Theory of games and economic behaviour.* Princeton University Press, Princeton, 1953. 3rd edition.

[37] B. von Stengel. LP representation and efficient computation of behavior strategies. Technical Report S-9301, University of the Federal Armed Forces at Munich, 1993.

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

[39] B. von Stengel. Computing equilibria for two-person games. In R. J. Aumann and S. Hart, editors, *Handbook of Game Theory with Economic Applications*, volume 3, chapter 45. North-Holland, 2002.

[40] B. von Stengel, A. van den Elzen, and A. J. J. Talman. Computing normal form perfect equilibria for extensive two-person games. *Econometrica*, 70:693–715, 2002.

[41] Robert B. Wilson. Computing simply stable equilibria. *Econometrica*, 60(5):1039–1070, 1992.