Computing Proper Equilibria of Zero-Sum Games

Peter Bro Miltersen and Troels Bjerre Sørensen

University of Aarhus, Denmark.

Abstract. We show that a *proper* equilibrium of a matrix game can be found in polynomial time by solving a linear (in the number of pure strategies of the two players) number of linear programs of roughly the same dimensions as the standard linear programs describing the Nash equilibria of the game.

1 Introduction

It has been known for more than fifty years that Nash equilibria of *matrix games* (i.e., two-player zero-sum games in normal form) coincide with pairs of maximin and minimax mixed strategies and can be found efficiently using linear programming. However, as is also well-established in game theory, the notion of a Nash equilibrium is too permissive to always prescribe sensible behavior. As an example, consider the classical example of *penny matching*: Alice has to guess whether Bob hides a penny heads or tails up. If she guesses correctly, she gets the penny. The payoff matrix for this game, with Alice being the row player trying to maximize payoff and Bob being the column player trying to minimize it, is as follows:

	hide penny heads up	hide penny tails up
guess "heads up"	1	0
guess "tails up"	0	1

It is well known and easy to see that the unique pair of maximin, minimax strategies and hence the unique Nash equilibrium in *penny matching* is for both players to mix up their two pure strategies uniformly, i.e., Alice guesses "heads up" with probability exactly $\frac{1}{2}$ and Bob hides the penny heads up with probability exactly $\frac{1}{2}$. Thus, the value of the game is $\frac{1}{2}$. There is not much more to say about this positively valued game, except that Bob clearly does not want to play it at all. Let us consider a modified version, *parsimonious penny matching*, where we give Bob the option of *teasing* Alice, by only pretending to hide a penny, but never really putting a penny at risk. This game is described by the following payoff matrix:

	hide penny heads up	hide penny tails up	tease
guess "heads up"	1	0	0
guess "tails up"	0	1	0

It is clear that the value of *parsimonious penny matching* is 0 and that Bob's unique minimax strategy is to "tease" with probability 1. It is more interesting to consider the situation for Alice: *Any* mix of her two pure strategies is a maximin strategy (for instance, guessing "heads up" with probability 1 is a maximin strategy). The reason is that to be a maximin strategy, it is sufficient to guarantee that Alice achieves the value of the game. This value is 0 and Alice will achieve this no matter what she does. Thus, any strategy profile in which Bob chooses "tease" with probability 1 is a Nash equilibrium. But only one of these prescribes sensible (in an intuitive sense) behavior for Alice: The one where she uniformly mixes her two pure strategies as she did in the unmodified penny matching game. Indeed, it seems that Alice ought to hope that Bob (non-sensibly) chooses to hide a penny after all. Just in case he does, she should opportunistically try to get as much as she can out of such a situation and play as she would in the unmodified game.

To formalize considerations such as the above (and similar considerations for much more intricate games, including general-sum ones), *refinements* of the Nash equilibrium concept have been considered. A particularly appealing one is Myerson's notion of a *proper* equilibrium [13]. An equilibrium 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 (i.e., a strategy for each player) 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 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. We refer to Myerson's paper for an intuitive justification of this refinement, but note for now that the unique proper equilibrium for *parsimonious penny matching* is the "sensible" strategy profile where Bob chooses "tease" with probability 1 and Alice uniformly mixes her two pure strategies.

In his seminal monograph on equilibrium refinements, van Damme [16], based on earlier work by Dresher [4], outlined a procedure for computing a proper equilibrium of a given matrix game. As he and Dresher describe the procedure, it is inherently exponential time. The main technical result of the present paper is a modification of the procedure so that it becomes a polynomial time algorithm. Thus, our main result is the following.

Theorem 1. *A proper equilibrium for a matrix game can be found in polynomial time in the size of the given payoff matrix.*

In addition, the algorithm we describe is also practical: We show that a proper equilibrium in a matrix game may be found by solving a linear (in the number of pure strategies of the two players) number of linear programs of roughly the same dimensions as the usual linear program describing the maximin/minimax strategies. Additional motivation for our result comes from Fiestras-Janeiro *et al* [5] who define a notion of properness for solutions of general linear programs (not necessarily describing optimal strategies for matrix games) by a reduction to the matrix game case and argue that this notion is relevant as a solution concept for general linear programs. They restate the exponential procedure of Dresher and van Damme as a way of finding the proper solutions. By the algorithm we present, we also get a polynomial time algorithm for finding a proper solution in the sense of their paper for general linear programs.

In the rest of the paper, we present our efficient algorithm for finding proper equilibria of matrix games. In Section 2, we present the original Dresher procedure which was shown by van Damme to find a proper equilibrium for a matrix game. Even though the philosophical motivation of Myerson's notion of a proper equilibrium is beyond the scope of this paper, the reader unfamiliar with Myerson's notion should be able to intuitively see why the equilibrium Dresher's procedure finds is "sensible". In Section 3, we present our efficient modification and give an example of its execution. In Section 4, we conclude with a discussion on the relevance of our algorithm for AI applications and in particular, we ask if it can be extended to solving games in extensive form (i.e., game trees).

2 Background

We review some notation from van Damme [16]. A matrix game is a 3-tuple $\Gamma = (\Phi_1, \Phi_2, R)$, where Φ_i is a finite nonempty set and R is a mapping $R : \Phi_1 \times \Phi_2 \to \mathbb{R}$. The set Φ_i is the set of *pure strategies* of Player i and R is the payoff function for Player 1. We assume that the elements of Φ_i are numbered and, consequently, we will speak about the k^{th} pure strategy of Player i. A mixed strategy s_i of Player i is a probability distribution on Φ_i. We denote the probability which s_i assigns to pure strategy k of Player i by s_i^k and we write S_i for the set of all mixed strategies of this player. If $s_i \in S_i$, then $C(s_i)$ denotes the *carrier* of s_i, i.e. $C(s_i) := \{k \in \Phi_i; s_i^k > 0.\}$ We denote the set of pure strategies which are in the carrier of some equilibrium strategy of Player i in Γ by $C_i(\Gamma)$. Note that $\Phi_i \backslash C_i(\Gamma)$ are those pure strategies for Player i that have probability 0 of being played in *all* equilibria. Thus, we can think of the strategies in $\Phi_i \backslash C_i(\Gamma)$ as the *superfluous* strategies. An alternative characterization is given below.

The payoff function R extends to mixed strategies by letting it denote the expected payoff when the mixed strategies are played. All equilibria of a matrix game Γ yield the same payoff to Player 1, and we denote this value $v(\Gamma)$ and call it the value of the game. We define $O_1(\Gamma) = \{s_1 \in S_1; R(s_1, l) \geqq v(\Gamma) \forall l \in \Phi_2\}$, and $O_2(\Gamma) = \{s_2 \in S_2; R(k, s_2) \leqq v(\Gamma) \forall k \in \Phi_1\}$. The set $O_i(\Gamma)$ is a convex polyhedron, the elements of which are called the *optimal strategies* of Player i in Γ. $O_1(\Gamma)$ and $v(\Gamma)$ can be determined by solving the linear programming problem (1):

$$\text{maximize}_{v, s_1 \in S_1}\ v \quad \text{s.t. } R(s_1, l) \geqq v \quad \text{for all } l \in \Phi_2 \tag{1}$$

It was shown by Bohnenblust, Karlin and Shapley [3], and by Gale and Sherman [6] that $\Phi_2 \backslash C_2(\Gamma)$ consists of exactly those pure strategies k so that for some mixed strategy $s_1 \in O_1$ we have that $R(s_1, k) > v(\Gamma)$. Thus, we can also think of $\Phi_2 \backslash C_2(\Gamma)$ as the *exploitable* pure strategies for Player 2: The strategies for Player 2 against which it is possible for Player 1 to play optimally, yet get more than his "fair share" of the game. Both characterizations of $\Phi_2 \backslash C_2(\Gamma)$ will be useful below.

We have now introduced the relevant notation to understand van Damme's reformulation [16, page 59] of the original procedure due to Dresher:

> For a matrix game $\Gamma = (\Phi_1, \Phi_2, R)$ *Dresher's procedure* for selecting a particular optimal strategy of player 1 is described as follows:
>
> (i) Set $t := 0$, write $\Phi_1^t := \Phi_1$, $\Phi_2^t := \Phi_2$ and $\Gamma^t := (\Phi_1^t, \Phi_2^t, R)$. Compute $O_1(\Gamma^t)$, i.e. the set of optimal strategies of player 1 in the game Γ^t.
>
> (ii) If all elements of $O_1(\Gamma^t)$ are equivalent in Γ^t, then go to (v), otherwise go to (iii).
>
> (iii) Assume that player 2 makes a mistake in Γ^t, i.e. that he assigns a positive probability only to the pure strategies which yield player 1 a payoff greater than $v(\Gamma^t)$. Hence, restrict player 2's pure strategies set to $\Phi_2^t \backslash C_2(\Gamma^t)$.
>
> (iv) Determine the optimal strategies of player 1 which maximize the minimum gain resulting from mistakes of player 2. Hence, compute the optimal strategies of player 1 in the game $\Gamma^{t+1} := (\Phi_1^{t+1}, \Phi_2^{t+1}, R)$, where $\Phi_1^{t+1} := ext\ O_1(\Gamma^t)$ is the (finite) set of extreme optimal strategies of player 1 in Γ^t and $\Phi_2^{t+1} := \Phi_2^t \backslash C_2(\Gamma^t)$. Replace t by $t+1$ and repeat step (ii).
>
> (v) The set of *Dresher-optimal* (or shortly *D-optimal strategies*) of player 1 in Γ is the set $D_1(\Gamma) := O_1(\Gamma^t)$.

It was shown by van Damme that if the above procedure is used to find a D-optimal strategy for Player 1 and the analogous procedure is used to find a D-optimal strategy for Player 2, then the strategy profile resulting from combining them is a proper equilibrium in the sense of Myerson.

We now discuss how to interpret the algorithm and analyze the implications for its complexity. First, strictly speaking, $O_1(\Gamma^t)$ is a set of mixed strategies for Player 1 in the game Γ^t, not the game Γ, so we need to understand how to interpret line (v). However, as is clear from the procedure, each pure strategy for Player 1 in Γ^i corresponds to a mixed strategy for Player 1 in Γ^{i-1}. Thus, each mixed strategy in Γ^i also corresponds to a mixed strategy in Γ^{i-1} and by iterating this interpretation, each mixed strategy in Γ^t can also be interpreted as a mixed strategy in Γ. In the following section, it will be convenient to have some notation for this interpretation: If s is a mixed strategy for Player 1 in Γ^i for some i, we let $\widehat{s}$ be the corresponding mixed strategy in Γ and also extend this notation to sets of strategies. Second, it is not quite clear what is meant by "Compute $O_1(\Gamma^t)$" in line (i), i.e., what representation is intended at this point in the procedure for this infinite object. However, in line (iv) we are going to need ext $O_1(\Gamma^t)$, i.e., the set of all corners of $O_1(\Gamma^t)$, so we can assume that this is the finite representation we use. Indeed, van Damme is very explicit that the set of corners of the polytope $O_1(\Gamma^t)$ will be explicitly computed: He refers in the text following the procedure to an algorithm by Balinski [1] for performing such an enumeration and he notes that there are a finite number of extreme points. Also in Dresher's original formulation is it very clear that an enumeration is to be performed, and Dresher even carries out such an enumeration for a small example. *This explicit enumeration is the main source of inefficiency of the algorithm.* Indeed, it is well known and easy to see that in the worst case, the number of extremal points of a polytope defined by a linear program is exponential in the size of the linear program. Thus, the Dresher procedure as stated is an exponential procedure in the worst case sense. Also, in practice, enumerating all extremal optimal solutions to a linear program (even when this set is small) is a much more elaborate process than just finding an optimal solution. Finally, it is not explicitly stated by van Damme how to compute $\Phi_2^{t+1} := \Phi_2^t \backslash C_2(\Gamma^t)$ in line (iv) of the algorithm. In the original version by Dresher, it is done by letting $C_2(\Gamma^t)$ be the subset of Φ_2^t which yield the value of the game against every optimal strategy of Player 1, i.e., by using the characterization of $\Phi_2^t \backslash C_2(\Gamma^t)$ as the exploitable strategies for Player 2. As we have an explicit representation of ext $O_1(\Gamma^t)$, and it is enough to check for optimality against this finite set, this is one possibility. Another way to do compute $C_2(\Gamma^t)$, which is not very practical but at least polynomial, is to check each of $k \in \Phi_2^t$ for membership of $C_2(\Gamma^t)$. This could be done by solving $|\Phi_2^t|$ linear programs of the following form:

$$\begin{aligned} &\max_{x,p} \; p \\ \text{s.t.} \quad &{A'}^{\top} x \geq e_k p \\ &x^{\top} \mathbf{1}_m = 1 \\ &x \geq \mathbf{0}_m \end{aligned}$$

where e_k is the kth standard basis vector, and $\mathbf{0}_i$ and $\mathbf{1}_i$ are constant column vectors of height i, filled with 0s and 1s respectively, and A' is the $m \times |\Phi_2^t|$ payoff matrix of the game Γ^t with the *value* of the game Γ^t subtracted from each entry (i.e., the game matrix is "normalized" so that it has value zero). An optimal solution to the linear program with a positive value of p corresponds to an optimal strategy for Player 1 obtaining payoff strictly larger than v against the k'th pure strategy of Player 2, i.e, we have that k is exploitable by the characterization of Gale and Sherman mentioned above. This is the case if and only if k is not in $C_2(\Gamma^t)$. Alternatively, we could write a linear program whose set of feasible solutions is the optimal mixed strategies for Player 2 and with the objective function to maximize being the probability of choosing k. This formulation directly expresses whether k is superfluous or not. In both cases, we should solve $|\Phi_2^t|$ linear programs in the t'th iteration of the procedure, leading to a worst case *quadratic* number of programs being solved in total during the execution of Dresher's procedure.

3 Algorithm

To improve on the efficiency of Dresher's procedure, we have to change the way $O_1(\Gamma^t)$ is represented, since we can not afford to enumerate the extreme points of this polyhedron. Since $O_1(\Gamma^t)$ is the set of optimal solutions to a linear program, it can be represented as a set of linear constraints. Our approach is to include the linear constraints of $O_1(\Gamma^{t-1})$ in the linear program used to obtain $O_1(\Gamma^t)$, or actually $\widehat{O_1(\Gamma^t)}$, i.e., the corresponding set of mixed strategies in the original game.

Lemma 2. *For all t, the set $\widehat{O_1(\Gamma^t)}$ is the set of x^*-parts and the value of the game Γ^t is the z^*-part of optimal solutions (x^*, z^*) to the LP:*

$$
\begin{aligned}
P_t : \qquad & \max_{x\in\mathbb{R}^m,\ z\in\mathbb{R}} \ z \\
\text{s.t.} \quad & {A'_i}^\top x \geq \mathbf{0}_{n'_i}, \quad \forall i : 0 \leq i < t \\
& {A_t}^\top x \geq \mathbf{1}_{n_t} z \\
& x^\top \mathbf{1}_m = 1 \\
& x \geq \mathbf{0}_m
\end{aligned}
$$

where m is $|\Phi_1|$, n_i is $|\Phi_2^i|$ and n'_i is $|C_2(\Gamma^i)|$, A'_i is the $m \times n'_i$ payoff matrix of the game $\Upsilon'_i = (\Phi_1, C_2(\Gamma^i), R)$ with the value of Γ^i (computed in a previous round) subtracted from each entry, and A_t is the $m \times n_t$ payoff matrix of the game $\Upsilon_t = (\Phi_1, \Phi_2^t, R)$.

The above lemma gives us an alternative way of computing $O_1(\Gamma^t)$. We next present an alternative way of computing $C_2(\Gamma^t)$.

Lemma 3. *Player 2's superfluous strategies in Γ^t, i.e., $\Phi_2^t \backslash C_2(\Gamma^t)$, are those k such that $p_k = 1$ in any (and all) optimal solutions to the LP:*

$$
\begin{aligned}
Q_t : \qquad & \max_{x\in\mathbb{R}^m,\ p\in\mathbb{R}^{n_t}} \ p^\top \mathbf{1}_{n_t} \\
\text{s.t.} \quad & {A'_i}^\top x \geq \mathbf{0}_{n'_i}, \quad \forall i : 0 \leq i < t \\
& {A''_t}^\top x \geq p \\
& p \leq \mathbf{1}_{n_t} \\
& p \geq \mathbf{0}_{n_t} \\
& x \geq \mathbf{0}_m
\end{aligned}
$$

with the same definitions as in Lemma 2 and with A''_t being A_t with the value of Γ^t (found when solving P_t), subtracted from each entry.

Due to the space constraints, we omit the proofs of the lemmas. We are now ready to state our modification of Dresher's procedure:

Modified Dresher procedure

(i) Set $t := 0$, and let $\Phi_2^0 := \Phi_2$.
(ii) Find an optimal solution to P_t.

(iii) Find an optimal solution to Q_t. Let Φ_2^{t+1} be those $k \in \Phi_2^t$ where $p_k = 1$ in the optimal solution found.
(iv) If $\Phi_2^{t+1} = \emptyset$ then go to (v) else replace t by $t+1$ and go to (ii).
(v) The set of D-optimal strategies of Player 1 in Γ is the set of optimal solutions to P_t. Output any one of these optimal solutions.

Lemma 2 and 3 give us that the optimal solutions to P_t for the terminal value of t are indeed the D-optimal strategies for Player 1. By computing a D-optimal strategy for Player 1 and afterwards a D-optimal strategy for Player 2 by applying the procedure a second time, we have computed a proper equilibrium.

That the above given procedure runs in polynomial time, can be seen by observing that $|\Phi_2^t|$ decreases by at least 1 in each iteration. This means that we solve at most $|\Phi_2|$ linear programs of the form of P_t and just as many of the form of Q_t. The number of variables in P_t is $|\Phi_1| + 1$, and the number of constraints is $|\sum_{i=0}^{t-1} C_2(\Gamma^i)| + |\Phi_2^t| + 1 = |\sum_{i=0}^{t-1} C_2(\Gamma^i)| + |\Phi_2 \setminus \bigcup_{i=0}^{t-1} C_2(\Gamma^i)| + 1 = |\sum_{i=0}^{t-1} C_2(\Gamma^i)| - |\sum_{i=0}^{t-1} C_2(\Gamma^i)| + |\Phi_2| + 1 = |\Phi_2| + 1$. This is independent of t, and it is also the same number of constraints used to find just a Nash equilibrium in the standard way, i.e., the number of constraints in the linear program (1). The number of variables in Q_t is $|\Phi_1| + |\Phi_2^t|$, which is less than $|\Phi_1| + |\Phi_2|$ for all t. The number of constraints is the same as in P_t, not counting simple bounds on variables. We thus solve at most a linear number of linear programs of sizes comparable to the size of the linear program (1). Since linear programs are polynomial time solvable, the entire procedure is polynomial time. Also, from a more practical point of view, Notice that an optimal solution to P_i is a feasible solution to P_{i+1}, allowing us to "warm start" an LP-solver on P_{i+1}. Notice as well that the x-part of an optimal solution to P_i is a feasible solution to Q_i when the remaining variables are set to 0, again allowing for a "warm start".

3.1 Example

As an example of an execution of the algorithm, we will now find the proper strategy for Alice in the game of *parsimonious penny matching* from the introduction. The first linear program we need to solve, P_0 is the usual linear program for finding the Nash equilibria of the game.

$$\begin{aligned} P_0 : \quad & \max_{x,z} \; z \\ \text{s.t.} \quad & 1x_1 + 0x_2 \geq z \\ & 0x_1 + 1x_2 \geq z \\ & 0x_1 + 0x_2 \geq z \\ & x_1 + x_2 = 1 \\ & x_1, x_2 \geq 0 \end{aligned}$$

Solving this, we find that the value of the game is $z^* = 0$. The next step is to decide which of Bob's strategies are superfluous. This is done by solving Q_0.

Since z^* was 0, A''_0 is equal to A.

$$\begin{aligned} Q_0: \quad & \max_{x,p} \; p_1 + p_2 + p_3 \\ \text{s.t.} \quad & 1x_1 + 0x_2 \geq p_1 \\ & 0x_1 + 1x_2 \geq p_2 \\ & 0x_1 + 0x_2 \geq p_3 \\ & p_1, p_2, p_3 \leq 1 \\ & p_1, p_2, p_3 \geq 0 \\ & x_1, x_2 \geq 0 \end{aligned}$$

Solving this, we find an optimal solution $x^* = [1, 1]^\top, p^* = [1, 1, 0]^\top$, and therefore conclude that Bob's two first strategies are superfluous, i.e. that he would not willingly hide a penny. In the next iteration, Alice refines her strategy, trying to gain as much as possible from a mistake of Bob, while maintaining optimality in case no such mistake is made. Thus, we solve P_1:

$$\begin{aligned} P_1: \quad & \max_{x,z} \; z \\ \text{s.t.} \quad & 0x_1 + 0x_2 \geq 0 \\ & 1x_1 + 0x_2 \geq z \\ & 0x_1 + 1x_2 \geq z \\ & x_1 + x_2 = 1 \\ & x_1, x_2 \geq 0 \end{aligned}$$

The unique solution is $x^* = [\frac{1}{2}, \frac{1}{2}]^\top, z^* = \frac{1}{2}$. Thus, Alice can expect to gain half a penny if Bob makes the mistake of not teasing. We then check whether we can refine the strategy even further by solving Q_1:

$$\begin{aligned} Q_1: \quad & \max_{x,p} \; p_1 + p_2 \\ \text{s.t.} \quad & 0x_1 + 0x_2 \geq 0 \\ & \tfrac{1}{2}x_1 - \tfrac{1}{2}x_2 \geq p_1 \\ & -\tfrac{1}{2}x_1 + \tfrac{1}{2}x_2 \geq p_2 \\ & p_1, p_2 \leq 1 \\ & p_1, p_2 \geq 0 \\ & x_1, x_2 \geq 0 \end{aligned}$$

The optimal solution has $p^* = [0, 0]^\top$, and thus there are no further mistakes that can be exploited.

4 Discussion

Our main result deals with finding proper equilibria in zero-sum *normal form* games, i.e. games given by a payoff matrix. However, in many realistic situations where it is desired to compute prescriptive strategies for games with hidden information, in particular, the kinds of strategic games considered by the AI community, the game is given in *extensive form*. That is, the game is given 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 [8] 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. The minimax strategies can be found from the sequence form by solving a linear program of size linear in the size of the game tree, avoiding the conversion to normal form altogether.

The Koller-Megiddo-von Stengel algorithm has been used by the AI community for solving many games, in particular variants of poker, some of them very large [14,2,7]. However, as was first pointed out by Koller and Pfeffer [9], the equilibria computed by the Koller-Megiddo-von Stengel procedure may in general be "non-sensible" in a similar sense as discussed above for matrix games. Alex Selby [14], computing a strategy for a variant of Hold'Em poker found similar problems. In a recent paper [12], we suggested that the notion of equilibrium refinements from game theory would be a natural vehicle for sorting out the insensible equilibria from the sensible ones, also for the application of computing prescriptive strategies for extensive-form zero-sum games, to be used by game playing software. We showed how to modify the Koller-Megiddo-von Stengel algorithm so that a *quasi-perfect* equilibrium (an equilibrium refinement due to van Damme [15]) is computed, and we showed how computing such an equilibrium would eliminate the insensible behavior in the computed strategy alluded to in Selby's poker example and in many other examples as well.

An equilibrium for a zero-sum extensive form game is said to be *normal-form proper* if the corresponding equilibrium for the corresponding matrix game is proper. It was shown by van Damme that normal-form properness is a further refinement of quasi-perfection. Here, we show an example of an equilibrium for a fairly natural extensive-form game we call *Penny matching on Christmas morning*. The equilibrium arguably prescribes insensible play. However, it is quasi-perfect, and in fact, the algorithm of [12] gives the insensible equilibrium as output. However, the equilibrium is not normal-form proper, thus suggesting that this further refinement is also relevant for prescribing proper play in extensive-form zero-sum games. The game of Penny matching on Christmas morning is as follows. Recall from the introduction that in the standard penny matching game, Bob (Player 2) hides a penny and Alice (Player 1) 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

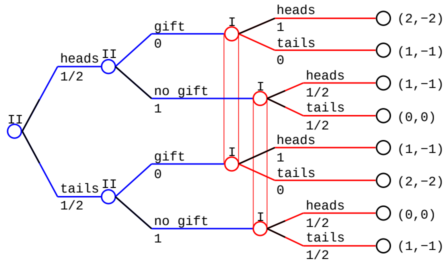

Fig. 1. *Penny matching on Christmas morning* - "bad" equilibrium

extensive form of this game as well as the pair of maximin/minimax behavioral strategies computed by the game theory software tool Gambit [11] using the Koller-Megiddo-von Stengel algorithm is given in Figure 1. We see that if Player 1 does not receive a gift, the strategy computed 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 computed suggests that she guesses heads with probability 1. This does not seem sensible. Indeed, if she had randomized her guess, as in the "no-gift" scenario, her conditional expected payoff, conditioned by the fact that she receives the gift, would be guaranteed to be at least a penny and a half. On the other hand, with the strategy suggested, this conditional expected payoff is only a penny in the case where the strategy of Player 2 happens to be the pure strategy of hiding the penny tails up and giving the gift. Thus, it seems that the unique sensible equilibrium for the game is the one where Player 1 randomizes her guess uniformly, even after having received a gift.

The "bad" equilibrium is quasi-perfect and a possible output of the algorithm for computing quasi-perfect equilibria of [12]. However, it is not normal-form proper and in fact the unique normal-form proper equilibrium for the game is the "good" equilibrium where player 1 randomizes her guess uniformly, even after having received a gift. This can be seen by converting the game to normal form and applying either the original Dresher's procedure or the version from this paper. We are not aware of any other equilibrium refinement notion that handles this and similar examples "correctly". It thus seems quite motivated to study methods for computing a normal-form proper equilibrium for a given extensive-form zero-sum game. We may do this by converting the game into normal form (incurring an exponential blowup in the size of the representation) and running Dresher's procedure. If the original version of Dresher's procedure were used, we would have a doubly-exponential time procedure. If the version of Dresher's procedure suggested in this paper is used, we have a singly-exponential time procedure. Ideally, we would like some way of combining the Koller-Megiddo-von Stengel algorithm with Dresher's procedure and obtain a polynomial time procedure, but don't see an obvious way of doing this. We thus leave the following as a major open problem:

Open Problem 1 *Can a normal-form proper equilibrium of an extensive-form two-player zero-sum game with perfect recall be found in time polynomial in the size of the given extensive form?*

It is interesting to note that insisting on normal-form properness provides an intriguing and non-trivial solution to the problem of choosing between different minimax strategies even in *perfect information* games, a problem recently studied by Lorenz [10] using an approach very different from the equilibrium refinement approach. As an example, consider the game given in Figure 2 (payoffs are paid by Player 2 to Player 1). The value of the game for Player 1 is 0 and he is

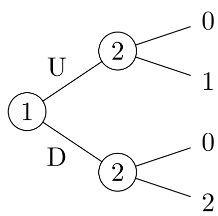

Fig. 2. *Up or Down?*

guaranteed to obtain this value no matter what he does. However, if he chooses U and his opponent makes a mistake, he will receive a payoff of 1. On the other hand, if he chooses D and his opponent makes a mistake, he will receive a payoff of 2. In the unique normal-form proper equilibrium for this game, Player I chooses U with probability 2/3 and D with probability 1/3 as can be seen by converting the program to normal form and applying Dresher's procedure. An intuitive justification for this strategy is as follows. Player 1 should imagine being up against a Player 2 that cannot avoid sometimes making mistakes, as otherwise the choice of Player 1 is irrelevant. On the other hand, Player 1 should assume that Player 2 is still a rational player who can make an effort to avoid making mistakes, and in particular train himself to avoid making mistakes in certain (but not all) situations. Thus, Player 1's strategy shouldn't be pure: In particular, if he chooses D with probability 1 (as is surely tempting), Player 2 may respond by concentrating his efforts to avoid making mistakes in his bottom node. Then, Player 1 will not get his "fair share" out of Player 2's mistakes. In conclusion, computing normal-form proper equilibria for zero-sum extensive-form games seems very interesting, even in the special case of perfect information games. Doing this special case efficiently might be easier than solving the general open problem above. It would also be interesting to compare this approach of selecting between different minimax solutions for such games with the very different approach of Lorenz.

References

1. M. Balinski. An algorithm for finding all vertices of convex polyhedral sets. *Journal of the Society for Industrial and Applied Mathematics*, 9(1):72–88, March 1961.
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 *International Joint Conference on Artificial Intelligence*, 2003.
3. H. F. Bohnenblust, S. Karlin, and L. S. Shapley. Solutions of discrete, two-person games. *Annals of Mathematical Studies*, pages 37–49, 1950.
4. Melvin Dresher. *The Mathematics of Games of Strategy: Theory and Applications*. Prentice-Hall, 1961.
5. M. G. Fiestra-Janeiro, I. Garcia-Jurado, and J. Puerto. The concept of proper solution in linear programming. *Journal of Optimization Theory and Applications*, 106(3):511–525, September 2000.
6. D. Gale and S. Sherman. Solutions of finite two-person games. *Annals of Mathematical Studies*, pages 37–49, 1950.
7. Andrew Gilpin and Tuomas Sandholm. Finding equilibria in large sequential games of incomplete information. Technical Report CMU-CS-05-158, Carnegie Mellon University, 2005.
8. 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.
9. Daphne Koller and Avi Pfeffer. Representations and solutions for game-theoretic problems. *Artificial Intelligence*, 94(1–2):167–215, 1997.
10. Ulf Lorenz. Beyond optimal play in two-person-zerosum games. *Lecture Notes in Computer Science*, 3221:749–759, January 2004.
11. 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.
12. 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*, Miami, Florida, January 2006. ACM-SIAM.
13. R. B. Myerson. Refinements of the Nash equilibrium concept. *International Journal of Game Theory*, 15:133–154, 1978.
14. Alex Selby. Optimal heads-up preflop holdem. Webpage, http://www.archduke.demon.co.uk/ simplex/index.html, 1999.
15. 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.
16. Eric van Damme. *Stability and Perfection of Nash Equlibria*. Springer-Verlag, 2nd edition, 1991.