

# Computing a Proper Equilibrium of a Bimatrix Game

TROELS BJERRE SØRENSEN, University of Warwick

We provide the first pivoting-type algorithm that computes an exact *proper* equilibrium of a bimatrix game. This is achieved by using Lemke’s algorithm to solve a linear complementarity problem (LCP) of polynomial size. This also proves that computing a simple refinement of proper equilibria for bimatrix game is PPAD-complete. The algorithm also computes a witness in the form of a parameterized strategy that is an  $\varepsilon$ -proper equilibrium for any given sufficiently small  $\varepsilon$ , allowing polynomial-time verification of the properties of the refined equilibrium. The same technique can be applied to matrix games (two-player zero-sum), thereby computing a parameterized  $\varepsilon$ -proper strategy in polynomial time using linear programming.

Categories and Subject Descriptors: F.2.2 [**Theory of Computation**]: Analysis of Algorithms and Problem Complexity – Nonnumerical Algorithms and Problems

General Terms: Algorithms, Economics, Theory

Additional Key Words and Phrases: Equilibrium Computation, Equilibrium Refinements, Proper Equilibrium

## 1. INTRODUCTION

Since the birth of game theory, a lot of work has gone into describing and computing *good* solutions to games. Most of this work has been focused on Nash equilibria. However, it has long been known [Selten 1965] that a strategy profile being a Nash equilibrium only implies stability in the sense that no player has an incentive to deviate; some of the stable strategy combinations may be undesirable, e.g., if a player is using dominated strategies. A lot of work has therefore gone into developing several [Selten 1975; Myerson 1978; Kreps and Wilson 1982; van Damme 1984] *refinements* of the Nash equilibrium, with the intent of eliminating such unreasonable equilibria. For a comprehensive account, see the monograph of van Damme [van Damme 1991].

The equilibrium refinement that we consider in this paper is the *proper equilibrium* of Myerson [Myerson 1978]. A bimatrix game (or two-player strategic form game), is given as two matrices  $A, B \in \mathbb{R}^{m \times n}$ . The game is played by the two players simultaneously choosing a pure strategy; Player 1 from the  $m$  rows, Player 2 from the  $n$  columns. Player 1 then receives payoff  $A_{i,j}$  and Player 2 receives payoff  $B_{i,j}$ , where  $i$  was the pure strategy chosen by Player 1 and  $j$  was the pure strategy chosen by Player 2. The players are also allowed to use mixed strategies, which are probability distributions over indices. If the strategy played by Player 1 is  $x \in \Delta^m$  and the strategy played by Player 2 is  $y \in \Delta^n$ , the respective expected payoff to each player is given by  $x^\top Ay$  and  $x^\top By$ .

---

Research supported by EPSRC award EP/G069034/1 and by the Centre for Discrete Mathematics and its Applications (DIMAP).

Author’s address: T. B. Sørensen, Department of Computer Science, University of Warwick, United Kingdom. Email: trolld@dcs.warwick.ac.uk.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org.

EC’12, June 4–8, 2012, Valencia, Spain.

Copyright 2012 ACM 978-1-4503-1415-2/12/06...\$10.00.

An equilibrium in mixed strategies is said to be proper, if it is a limit point of a sequence of  $\varepsilon$ -proper strategy profiles for  $\varepsilon \rightarrow 0^+$ . A strategy profile  $(x, y)$  is  $\varepsilon$ -proper if the following properties are satisfied:

$$\begin{aligned} \forall i, j \in [m] : (Ay)_i < (Ay)_j &\Rightarrow x_i \leq \varepsilon \cdot x_j \\ \forall i, j \in [n] : (x^\top B)_i < (x^\top B)_j &\Rightarrow y_i \leq \varepsilon \cdot y_j \\ \forall i \in [m] : x_i > 0 \\ \forall j \in [n] : y_j > 0 \end{aligned} \tag{1}$$

In other words, for each player, if pure strategy  $i$  performs worse than pure strategy  $j$  against the opponent's mixed strategy, then the strategy must assign a probability to pure strategy  $i$  that is infinitesimally smaller than the probability it assigns to pure strategy  $j$ , though both must still be strictly positive. The intention of the definition is to capture the behavior of players that are assumed to sometimes make mistakes, but with negligible probability and *in a rational manner*: The assumption is that players will make bigger mistakes with significantly smaller probability than the smaller mistakes, but that any mistake is played with infinitesimally small probability.

As an example of this, examine the game of Fig. 1. In any Nash equilibrium, the column player will always play  $a$ ; playing any other strategy would be a mistake, which could be exploited by the row player. The row player is unrestricted by Nash, but he can do better against a column player who sometimes makes mistakes. The row player can try to get as much out of a potential mistake of the column player as possible. By playing  $U$  with probability  $2/3$  and  $D$  with probability  $1/3$ , any mistake of the opponent will give the row player an expected reward of at least  $2/3$ . The column player can make sure that a mistake will only cost an expected  $2/3$ , by making it twice as likely that he will make the mistake of playing  $c$  than that he will play  $b$ , while he will play  $d$  with an even smaller probability. A  $(3\varepsilon)$ -proper strategy profile is given by the polynomials in  $\varepsilon$  stated in the below and to the right of the game. The limit as  $\varepsilon \rightarrow 0$  gives the desired proper strategy for the row player. Furthermore, it can be seen that  $(2/3, 1/3)$  is the unique proper strategy for the row player.

Kohlberg and Mertens [Kohlberg and Mertens 1986] and van Damme [van Damme 1984, 1991] established a number of attractive properties of proper equilibria of strategic-form games, giving additional motivation for computing them. One of the very desirable properties proven is that if one takes an extensive-form game, converts it to strategic-form [Kuhn 1953], computes a proper equilibrium, and translates the computed strategy profile back to the extensive-form game, the strategy will

	II				
I		a	b	c	d
U	0	0	1	1	$\frac{2}{3}$
D	0	2	0	2	$\frac{1}{3}$

$1 - 3\varepsilon + \varepsilon^2 \quad \varepsilon - \varepsilon^2 \quad 2\varepsilon - \varepsilon^2 \quad \varepsilon^2$

Fig. 1. A zero-sum game with labeled strategies at the top and left, and a  $3\varepsilon$ -proper strategy profile to the bottom and right. The payoff shown is for the row player, namely  $A$ . The payoff to column player is given by  $B = -A$ .

be a *quasi-perfect* equilibrium. The quasi-perfect equilibrium was introduced by van Damme [van Damme 1984] to resolve the inconsistency between the perfect equilibria of an extensive-form game and the perfect equilibria of the corresponding strategic-form game. It refines Kreps and Wilson’s sequential equilibria [Kreps and Wilson 1982]. See [Miltersen and Sørensen 2006b] for additional discussion on quasi-perfect equilibria. In that respect, proper equilibria can be said to be able to reconstruct the structure that is lost when a game is transformed from extensive-form to strategic form.

In Myerson’s original paper [Myerson 1978], the proof of existence of proper equilibria is non-constructive, and the definition offers very little help on how to compute it. Fortunately, Kohlberg and Mertens [Kohlberg and Mertens 1986, Prop. 5] provide an intuitive and constructive proof of existence of  $\varepsilon$ -proper equilibria. For a given game  $G$  and  $\varepsilon > 0$ , their construction is a new game,  $G'$ , where the players have pure strategies corresponding to all the  $n!$  permutations of the coordinates of

$$\frac{1 - \varepsilon}{1 - \varepsilon^n} (1, \varepsilon, \dots, \varepsilon^{n-1})$$

for a player with  $n$  strategies. Strategies in  $G'$  can be translated to strategies of  $G$  in a straightforward manner, as a mixed strategy over mixed strategies is just a mixed strategy. Kohlberg and Mertens prove that any equilibrium of  $G'$  will be an  $\varepsilon$ -proper equilibrium of  $G$ . This directly gives a naïve algorithm for computing an  $\varepsilon$ -proper equilibrium, for a given fixed  $\varepsilon$ : explicitly construct  $G'$  and compute an equilibrium using existing techniques. This, however, has several drawbacks, keeping in mind that the final goal is to compute an exact proper equilibrium. Firstly, the game  $G'$  is exponentially larger than  $G$ , since a player with  $n$  pure strategies in  $G$  will have  $n!$  strategies in  $G'$ . This alone makes it infeasible to solve, and the construction would not help in getting PPA membership for the two-player case. Secondly, the parameter  $\varepsilon$  has to be chosen small enough to ensure that the computed equilibrium is close to an actual proper equilibrium of  $G$ . To the author’s best knowledge, it is unknown exactly how small such an epsilon would have to be, but most likely it would have to be exponentially small in the size of the game. Thirdly, even with a moderately sized  $\varepsilon$ , the  $\varepsilon^{n-1}$  used in the construction of  $G'$  will be a very small number. To ensure the desired property of the computed strategy profile,  $G'$  must either be solved exactly or to a precision much smaller than  $\varepsilon^{n-1}$ , which rules out most off-the-shelf solvers. Finally, any fixed positive epsilon would, at best, lead to something close to a proper equilibrium, at the very least by making all probabilities strictly positive.

For ease of notation, we will omit the  $\frac{1-\varepsilon}{1-\varepsilon^n}$  normalization factor in the rest of the paper. Since it is just a uniform scaling of the probabilities, this does not change anything in the game, other than scaling the payoff of the players, so the algorithm works both with and without the scaling. This means that the strategy space of the players in  $G'$  is permutations over  $(\varepsilon^0, \varepsilon^1, \dots, \varepsilon^{n-1})$ . This object is known in the operation research community as a *generalized permutahedron* over the set  $\{\varepsilon^{i-1} \mid i \in [n]\}$ . This object is completely characterized by the following inequalities:

$$P_n = \left\{ x \in \mathbb{R}^n : \begin{array}{l} \sum_{i \in S} x_i \geq \sum_{i=1}^{|S|} \varepsilon^{n-i} \quad \forall S \subseteq [n] \\ \sum_{i \in [n]} x_i = \sum_{i=1}^n \varepsilon^{n-i} \end{array} \right\} \quad (2)$$

This formulation restricts the strategy space to that of  $G'$ , but without explicitly enumerating the exponentially many extremal points. It does, however, have an exponential number of constraints. This problem was recently fixed by a clever construction of Goemans [Goemans 2009], who provided an *extended formulation* of the permutahedron. Goemans construction will be described in Section 3. Using Goemans construction, we are able to express equilibria of  $G'$  as solutions to a linear complementarity problem (LCP) with  $O(n \log n)$  rows and columns, described in detail in Section 4. The LCP can be solved using Lemke’s algorithm, which is described in Section 5. To avoid degeneracy, Lemke’s algorithm uses symbolic infinitesimal perturbations of the input LCP. This perturbation can be used to compute an equilibrium of  $G'$  with a symbolic infinitesimal  $\varepsilon$ , details of which is discussed in Section 5.2. By construction, any equilibrium of  $G'$  will be an  $\varepsilon$ -proper equilibrium of  $G$ . It is exactly this relationship we will be using in this paper. The rest of the paper deals with how to compute an equilibrium of  $G'$  with a symbolic  $\varepsilon$ . Doing so establishes our main theorem:

**THEOREM 1.1.** *A symbolic  $\varepsilon$ -proper equilibrium for a given bimatrix game can be computed by applying Lemke’s algorithm to an LCP of polynomial size.*

Computing a Nash equilibrium of a bimatrix game is PPAD-complete [Daskalakis et al. 2006; Chen and Deng 2006]. PPAD is a class of total function search problems, where a directed graph is implicitly encoded by predecessor and successor circuits. The problem is: given a node with a successor but no predecessor, find a different node that also has degree 1. Such a node is guaranteed to exist, since you are given a node with odd degree, so there must be at least one other. Since all proper equilibria are also Nash equilibria, computing a proper equilibrium is also PPAD-hard. Each pivot of Lemke’s algorithm can be implemented by a circuit, and the length of the encoding of the numbers needed for Lemke’s algorithm can be bounded by a polynomial in the entries of the payoff matrices. The last part follows from the fact that the values of the all variables of a given choice of basis can be computed by a matrix inversion. Combined with an orientation [Todd 1976] of Lemke’s algorithm, the problem of computing an equilibrium of the game  $G'$  becomes PPAD-complete. However, not all  $\varepsilon$ -proper equilibria of  $G$  are equilibria of some  $G'$ , so there may be some proper equilibria of  $G$  that cannot be found using the method outlined in this paper. As there are no polynomial time algorithm known for checking whether a given equilibrium of  $G$  is proper or not, we do not have a proof that computing a proper equilibrium is in PPAD, only that a refinement of it is:

**COROLLARY 1.2.** *The refinement of proper equilibria, corresponding to Kohlberg and Mertens’ proof of existence, is PPAD-complete to compute for a given bimatrix game.*

Before continuing with the contribution of this paper, we first review some related research.

## 2. RELATED RESEARCH

Computation of proper equilibria of bimatrix games has long posed a challenge, with infinitesimal  $\varepsilon$  being an important obstacle. Yamamoto [Yamamoto 1993] presented a numerical procedure for computing a proper equilibrium of an  $n$ -player game, using a explicit variable non-infinitesimal  $\varepsilon$ . 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.

In a recent paper, Belhaiza et.al. [Belhaiza et al. 2012] treated the computation of proper equilibria. Their primary contribution is an method for checking whether a given strategy profile is a proper equilibrium, by numerically solving a number of

mixed quadratic programs. Their proposed method for computing a proper equilibrium consists of enumerating all extremal perfect Nash equilibria, and for each of them checking whether it is proper. As they note, for some games, none of the extremal perfect Nash equilibria are proper. In this case, they incorrectly state [Belhaiza et al. 2012, page 301]: “Step 6. Else randomize on the strategy profiles of extreme perfect equilibria (belonging to the same Selten subset) closest to a sequence of  $\varepsilon$ -proper equilibria to find such a sequence analytically.” The paper does not say how this randomization is chosen, but their example 3.4 suggests the intended randomization is uniform over the extremal points of some maximal Selten subset. This will rarely work in general. As an example, observe the game in Fig. 1. There are two extremal perfect equilibria:  $((1, 0), (1, 0, 0, 0))$  and  $((0, 1), (1, 0, 0, 0))$ , and the two together is the unique maximal Selten subset. However, the uniform mix between them,  $((\frac{1}{2}, \frac{1}{2}), (1, 0, 0, 0))$ , is not proper. In fact, finding the right randomization over the extremal points is just as hard as finding the row-player’s strategy of some proper equilibrium from scratch. To see this, take any bimatrix game  $G$  where Player 1 does not have dominated strategies. Now add a new strategy to Player 2, which when played gives both players a fixed payoff that is higher than all other payoffs of the game. In this new game, all strategy profiles where Player 2 only plays his new strategy are perfect equilibria. However, the only proper equilibria of the game are those where the row player also plays a strategy that is part of some proper equilibrium of  $G$ . For this game, Belhaiza et.al.’s method would return a strategy profile, where the row player played uniform over his strategies, which will be incorrect in general.

All the previous work presented above work with a non-infinitesimal  $\varepsilon$ , which is not followed up by the necessary numerical analysis to ensure that one would not run into numerical problems. In the present paper, these problems are circumvented by using a symbolic infinitesimal  $\varepsilon$ . This is possible, because Lemke’s algorithm already has to keep track of a perturbation with powers of a symbolic infinitesimal to avoid degeneracy. This also means that the final output of the algorithm will be strategy profiles with probabilities that are formal polynomials in  $\varepsilon$ . That such symbolic  $\varepsilon$ -proper equilibria exist for bimatrix games was argued by Miltersen and Sørensen [Miltersen and Sørensen 2008], referring to results of Blume et al. [Blume et al. 1991].

For zero-sum games, like the one in the example from the introduction, Dresher [Dresher 1961] argued that the *optimal* strategies were those that survived a certain lexicographic application of the minimax condition. Van Damme [van Damme 1991] later proved that the set of proper equilibria of a zero-sum game is exactly the Cartesian product of the polytopes of Dresher’s optimal strategies. This is 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. These *Dresher optimal* strategies can be computed in polynomial time [Miltersen and Sørensen 2006a], even when the game is given in extensive form [Miltersen and Sørensen 2008].

In coalitional games, computing the nucleolus [Schmeidler 1969] corresponds to computing a proper equilibrium of the related zero-sum *excess game* [Potters and Tijds 1992].

### 3. EXTENDED FORMULATION OF THE PERMUTAHEDRON

In this section, we briefly review the construction of Goemans [Goemans 2009], allowing for a polynomial size representation of the permutahedron. An extended formulation is, generally speaking, a polytope with few facets in a higher-dimensional space, such that when it is projected down into the target subspace, it will have the desired (often exponentially many) constraints. In Goemans’ construction,  $O(n \log n)$

extra variables (dimensions) are used, and the projection is simply on the  $n$  first coordinates.

The construction for the  $n$ -dimensional permutahedron starts with any *sorting network* on  $n$  elements. A sorting network is a very simple model for oblivious sorting, yet the smallest ones are asymptotically optimal [Ajtai et al. 1983] for comparison based sorting. Fig. 2 shows an example of a sorting network on three elements, which we will use as the example for constructing a representation of a permutahedron in three dimensions. In sorting networks, the inputs are assigned to each of the wires at the left side, and the network is evaluated from left to right. Every time a gate between two wires is encountered, the elements are compared, and the smaller of the two elements is moved to the top wire and the larger to the bottom wire. If the sorting network is correct, the elements of the right-most wires will be sorted correctly, for any input to the network. In Goemans' construction, each piece of wire is assigned a variable (as in Fig. 2), where the  $n$  left-most ones are the variables we wish to restrict to be in the permutahedron. Each gate of the sorting network is turned into three constraints; each of the input variables must individually be greater than the top output variable, and the sum of the input variables must equal the sum of the output variables. Finally, each of the right-most variables must equal the elements of the generalized permutahedron in sorted order. One way of seeing the construction is as an exaggerated sorting of the variables: each gate must output sorted elements with the correct sum, and where the smallest output is at least as small than the smallest input. Each gate must sort, but is allowed to spread the values out. After this spreading sorting, the values must be the extremal coordinates. Another way to see the construction is as a right-to-left mixing of the extremal coordinates. Each gate ensures that the values on the left are convex combinations of the values on the right. After enough of these pairwise mixes, the wires can contain any convex combination of the extremal coordinates. For the network in Fig. 2, the corresponding constraints are:

$$\begin{array}{cccc}
 x_1 \geq x_4 & x_3 \geq x_6 & x_4 \geq x_7 & x_7 = \varepsilon^2 \\
 x_2 \geq x_4 & x_5 \geq x_6 & x_6 \geq x_7 & x_8 = \varepsilon^1 \\
 x_1 + x_2 = x_4 + x_5 & x_3 + x_5 = x_6 + x_9 & x_4 + x_6 = x_7 + x_8 & x_9 = \varepsilon^0
 \end{array} \quad (3)$$

For reasons of clarity, the construction in Goemans' paper is for the normal permutahedron, where the elements are  $\{1, \dots, n\}$ . However, it easily extends in the above manner for the generalized permutahedron [Goemans 2010].

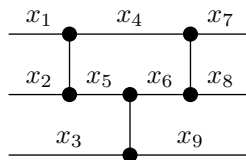


Fig. 2. A sorting network on three elements with labeled wires

#### 4. STRATEGY CONSTRAINTS AND LCP

The constraints derived from Goemans' construction can be neatly packed into strategy constraint matrices, which we will use of the rest of the paper. This allows us to use the general framework of Koller et.al. [Koller et al. 1996] for computing with strategy spaces given by linear constraints. To keep notation simple, we let  $x$  and  $y$  be the entire vector of the extended formulation, including the added variables. The payoff matrices

$$\begin{aligned}
E = F &= \begin{pmatrix} 1 & 1 & -1 & -1 & & & \\ & & 1 & & 1 & -1 & -1 \\ & & & 1 & & -1 & -1 \\ & & & & 1 & -1 & -1 \\ & & & & & 1 & \\ & & & & & & 1 \\ & & & & & & & 1 \end{pmatrix} & e = f = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \varepsilon^2 \\ \varepsilon^1 \\ \varepsilon^0 \end{pmatrix} \\
G = H &= \begin{pmatrix} -1 & & & & 1 & & & \\ & -1 & & & 1 & & & \\ & & -1 & & & 1 & & \\ & & & -1 & & 1 & & \\ & & & & -1 & & 1 & \\ & & & & & -1 & & 1 \end{pmatrix}
\end{aligned}$$

Fig. 3. Strategy constraint matrices for a  $3 \times 3$  game, using the sorting network from Fig. 2.

$A$  and  $B$  are padded with extra rows and columns of 0's to form the payoff matrices for the extended formulation. The valid strategies in  $G'$  for Player 1 are given by  $Ex = e$ ,  $Gx \leq 0$ , and  $x \geq 0$ , while the valid strategies for Player 2 are given by  $Fy = f$ ,  $Hy \leq 0$ , and  $y \geq 0$ . See Fig. 3 for an example of how these matrices could be constructed. The derivation follows the same line as [Koller et al. 1996], and goes as follows. Given a fixed strategy  $y$ , the best response  $x$  is an optimal solution to the linear program (dual on the right):

$$\begin{aligned}
&\underset{x}{\text{maximize}} && x^\top(Ay) && \underset{p, g}{\text{minimize}} && e^\top p \\
&\text{subject to} && x^\top E^\top = e^\top && \text{subject to} && E^\top p + G^\top g \geq Ay \\
&&& x^\top G^\top \leq 0 && && g \geq 0 \\
&&& x \geq 0 && &&
\end{aligned} \tag{4}$$

Feasible solutions  $x$  and  $(p, g)$  to these two LPs are optimal if and only if the objective values agree, i.e.,  $x^\top Ay = e^\top p$ . Using the first constraint of (4), this is equivalent to  $x^\top Ay = x^\top E^\top p$ . Combined with the first constraint of the dual in (4), this implies that

$$x^\top(-Ay + E^\top p + G^\top g) = 0 \tag{5}$$

This is the complementary slackness condition of the linear program, that states that the vector of slacks in the dual is orthogonal to the primal solution vector. Similarly,  $y$  is a best response to  $x$  iff it satisfies the strategy constraints

$$Fy = f, \quad Hy \leq 0 \tag{6}$$

and there exist vectors  $(q, h)$  that are feasible in the dual of the column players best-response LP

$$F^\top q + H^\top h \geq B^\top x \tag{7}$$

and satisfies the complementary slackness condition

$$y^\top(-B^\top x + F^\top q + H^\top h) = 0 \tag{8}$$

An LCP in standard form is specified by a pair  $(b, M)$  with  $b \in \mathbb{R}^n$  and  $M \in \mathbb{R}^{n \times n}$ . The problem is to find  $z \in \mathbb{R}^n$  such that

$$\begin{aligned} b + Mz &\geq 0 \\ z^\top(b + Mz) &= 0 \\ z &\geq 0 \end{aligned} \tag{9}$$

The constraints specified in this section can be packed into an LCP in standard form in the following way. If we let  $z = (x, y, p', p'', g, q', q'', h)^\top$ , where  $p = p' - p''$  and  $q = q' - q''$ , and

$$M = \begin{pmatrix} -A & E^\top & -E^\top & G^\top \\ -B^\top & & & F^\top & -F^\top & H^\top \\ -E & & & & & \\ E & & & & & \\ -G & & & & & \\ & -F & & & & \\ & F & & & & \\ & -H & & & & \end{pmatrix} \quad \text{and} \quad b = \begin{pmatrix} 0 \\ 0 \\ e \\ -e \\ 0 \\ f \\ -f \\ 0 \end{pmatrix}. \tag{10}$$

then  $z \geq 0$  and  $b + Mz \geq 0$  corresponds to the inequalities below on the left. Combining those with and  $z^\top(b + Mz) = 0$  imply the equalities below on the right:

$$\begin{aligned} -Ay + E^\top p + G^\top g &\geq 0 & x^\top(-Ay + E^\top p + G^\top g) &= 0 \\ -B^\top x + F^\top q + H^\top h &\geq 0 & y^\top(-B^\top x + F^\top q + H^\top h) &= 0 \\ -Ex + e &= 0 & g^\top(-Gx) &= 0 \\ -Gx &\geq 0 & h^\top(-Hx) &= 0 \\ -Fy + f &= 0 \\ -Hx &\geq 0 \\ x, y, g, h &\geq 0 \end{aligned} \tag{11}$$

As the complementary slackness conditions are enforced by the LCP,  $x$  and  $y$  are best responses to each other, and a solution to the above LCP is thus an equilibrium of  $G'$ .

## 5. LEMKE'S ALGORITHM

Lemke [Lemke 1965] provided an algorithm for solving LCPs given in the standard form, shown in (9). It works by relaxing the problem, by replacing  $(b + Mz)$  with  $(b + Mz + d \cdot z_0)$ , where  $z_0$  is a non-negative scalar variable, and  $d$  is a constant vector, known as the covering vector. For our purpose,  $d = (1, 1, \dots, 1)$ . This relaxation makes it easy to find an initial solution  $(w, z, z_0)$  to the relaxed problem

$$\begin{aligned} w &= b + Mz + d \cdot z_0 \\ z^\top w &= 0 \\ z, w &\geq 0 \end{aligned} \tag{12}$$

by choosing  $z_0$  large enough and  $z = 0$ . The algorithm performs a sequence of *complementary pivots*, searching for a solution with  $z_0 = 0$ . The pivots are similar to those



of the simplex method [Dantzig et al. 1955] for linear programming, but the entering variable is choosing so as to maintain the complementarity of  $z$  and  $w$ . Initially, the basis consists of the  $w$  variables, and all  $z$  variables are non-basic. The first entering variable is  $z_0$ . The first exiting variable is chosen such that all entries of  $w$  become non-negative. The exiting variable becoming 0 will allow the corresponding complementary variable to enter the basis. In all remaining iterations, the exiting variable is chosen by the minimal ratio test, and the entering variable as the corresponding complementary variable. This continues until  $z_0$  is evicted from the basis, in which case the solution is a feasible solution to the unrelaxed input problem. For a thorough exposition of Lemke's algorithm, see the monograph by Cottle et.al. [Cottle et al. 2009].

There are two problems with the algorithm as presented above. Primarily, there is an alternative undesirable way to terminate: the current entering variable might not be restricted by a basic variable. This is known as ray termination. To prove that the algorithm works for our case, we must prove that ray termination cannot happen. The second one is degeneracy: there may be multiple candidate variables to exit the basis. The latter one is handled through a slight perturbation of the  $b$  vector. It turns out that this solution also enables us to handle the infinitesimal  $\varepsilon$  that we need to correctly compute an equilibrium of  $G'$ . This will be discussed in Section 5.2.

### 5.1. Ray termination

To prove that ray termination cannot happen, we use the following theorem by Koller et.al. [Koller et al. 1996]:

**THEOREM 5.1 (KOLLER ET.AL.'96).** *If (i)  $z^\top Mz \geq 0$  for all  $z \geq 0$ , and (ii)  $z \geq 0$ ,  $Mz \geq 0$  and  $z^\top Mz = 0$  imply  $z^\top b \geq 0$ , then Lemke's algorithm computes a solution of the LCP (2.9) and does not terminate with a secondary ray.*

Our application of the theorem is similar to that of Koller et.al., and we need a similar set of lemmas.

**LEMMA 5.2.** *The only nonnegative solutions,  $x$  and  $y$ , to  $Ex = 0$ ,  $Gx \leq 0$ ,  $Fy = 0$ , and  $Hy \leq 0$ , are  $x = 0$  and  $y = 0$ .*

**PROOF.** This follows directly from the construction of the strategy constraint matrices from the sorting networks in the following way. Firstly, all the variables associated with output wires at the right end of the sorting network are forced to be 0, due to the last  $n$  rows of  $E$  and  $F$ . Secondly, each gate from right to left will have both output variables forced to be 0, forcing the sum of the input variables to be 0 as well. As each of the input variables must be greater than the smallest output variable, they must both be 0 as well. Thus,  $x = 0$  and  $y = 0$  is the only solution.  $\square$

**LEMMA 5.3.** *If  $E^\top p + G^\top g \geq 0$ ,  $g \geq 0$ ,  $F^\top q + H^\top h \geq 0$ , and  $h \geq 0$ , then  $e^\top p \geq 0$  and  $f^\top q \geq 0$ .*

**PROOF.** Consider the LP of the inequalities concerned with Player 1 (and its dual on the right):

$$\begin{array}{ll}
\underset{p,g}{\text{minimize}} & e^\top p \\
\text{subject to} & E^\top p + G^\top g \geq 0 \\
& g \geq 0
\end{array}
\qquad
\begin{array}{ll}
\underset{x}{\text{maximize}} & 0 \\
\text{subject to} & Ex = e \\
& Gx \leq 0 \\
& x \geq 0
\end{array}
\tag{13}$$

Since the dual is feasible and has value 0, by weak duality the objective function of the primal is lower bounded by 0, i.e.,  $e^\top p \geq 0$ . Similarly, for Player 2, we get  $f^\top q \geq 0$ .  $\square$

**THEOREM 5.4.** *If  $A \leq 0$  and  $B \leq 0$ , then  $M$  and  $b$  in (10) satisfy all assumptions of theorem 5.1.*

**PROOF.** Let  $z = (x, y, p', p'', g, q', q'', h)^\top \geq 0$ , and  $p = p' - p''$ ,  $q = q' - q''$ . Then we have that  $z^\top Mz = x^\top (-A - B)y \geq 0$ , satisfying condition (i) of theorem 5.1. Furthermore,  $Mz \geq 0$  implies  $-Ay + E^\top p + G^\top g \geq 0$ ,  $-B^\top x + F^\top q + H^\top h \geq 0$ ,  $Ex = e$ ,  $Gx \leq 0$ ,  $Fy = 0$ , and  $Hy \leq 0$ . Combining the latter with non-negativity of  $x$  and  $y$ , lemma 5.2 implies that  $x = 0$  and  $y = 0$ . Combining this with the first, we get  $E^\top p + G^\top g \geq 0$  and  $F^\top q + H^\top h \geq 0$ . By lemma 5.3, this implies that  $e^\top p \geq 0$  and  $f^\top q \geq 0$ . Finally,  $z^\top b = e^\top p + f^\top q \geq 0$ , showing that we satisfy condition (ii) of theorem 5.1.  $\square$

The conditions  $A \leq 0$  and  $B \leq 0$  can be ensured by subtracting a suitably large constant from the payoff of both players (but not from the rows and columns added by the extended formulation). This does not change the set of proper equilibria, as the value of best replies are shifted by the same amount. Thus, ray termination is not a possibility, and Lemke's algorithm will terminate with an equilibrium of  $G'$ .

## 5.2. Degeneracy and perturbation

The problem of degeneracy in LCPs is generally [Cottle et al. 2009] handled by perturbing the entries of the  $b$  vector by different powers of an infinitesimal value. Since this perturbation must not change the input so much as to change the set of solutions, the perturbation is kept as a symbolic infinitesimal. This means that the entries of  $b$ ,  $z$  and  $w$  will be formal polynomials in the symbolic infinitesimal. This does not cause problems for the algorithm, as the pivots only scales and adds these values. When a solution is found, normally only the 0th order term of the polynomials are used in the output, as the higher order terms are only used internally to avoid cycling. It is, however, not more complicated to output the solution as the full polynomials.

This feature of the algorithm also provides a way to compute an equilibrium of  $G'$  with a symbolic infinitesimal  $\varepsilon$ . The only places  $\varepsilon$  occurs in (10) is in the constant vector  $b$ , which is also to be perturbed to avoid degeneracy. This means that we can encode the powers of  $\varepsilon$  needed for the strategy constraints of  $G'$  directly into the perturbation in Lemke's algorithm. The only thing to keep in mind is that the  $n$  lowest powers of  $\varepsilon$  must be used for the strategy constraints, while higher powers can be used to avoid degeneracy. This means that we can read of the equilibrium of  $G'$  with symbolic  $\varepsilon$ , which means we will have a symbolic  $\varepsilon$ -proper equilibrium of  $G$  as required.

## 6. ZERO-SUM GAMES

The problem of computing proper equilibria is a lot easier for zero-sum games. Indeed, Dresher's procedure [Dresher 1961] already achieved this half a century ago by solving a sequence of linear programs. Nevertheless, by applying the above technique to zero-sum games, we can compute the set of proper equilibria using a single linear program in the following way. Given a proper equilibrium  $(x^*, y^*)$ , we can skip all the steps of Dresher's procedure, as the final output of the procedure will be  $\{x | x^\top A = x^{*\top} A\} \times \{y | Ay = Ay^*\}$ . Furthermore, we can use the computed symbolic  $\varepsilon$ -proper equilibrium to construct symbolic  $\varepsilon$ -proper equilibria for any given one of proper equilibria. This follows from van Damme's proof [van Damme 1991, Theorem 3.5.5] that Dresher optimal strategies are proper: given a convergent sequence of  $\varepsilon$ -proper equilibria, one can make it converge to any other payoff equivalent strategy profile.

For matrix games, the payoff to Player 2 is the negative of that to Player 1, i.e.,  $B = -A$ . This simplifies the computation, as we can use linear programming to compute equilibria. The derivation of the linear program is very similar to what was done for the bimatrix case. Notice that Player 2's strategy  $y$  only occurs on the constant right

hand side of dual in (4). Reintroducing  $y$  as a variable, and restricting it to be a valid strategy for Player 2 in  $G'$ , we get:

$$\begin{aligned}
& \underset{p, g, y}{\text{minimize}} && e^\top p \\
& \text{subject to} && E^\top p + G^\top g - Ay \geq 0 \\
& && Fy = f \\
& && Hy \leq 0 \\
& && g \geq 0
\end{aligned} \tag{14}$$

This linear program thus describes the payoff Player 2 can achieve, given that Player 1 plays a best response, and the optimal solutions are thus the minimax strategies for Player 2. A similar derivation for Player 1 leads to exactly the dual of (14) which describes the payoff Player 1 can achieve, against a Player 2 that plays best response, and thus also the minimax strategies for Player 1. Since these two are dual to each other, they also correspond to equilibria of  $G'$ .

While normal linear programs can be solved in polynomial time [Khachiyan 1979; Karmarkar 1984], we need to solve one with an infinitesimal  $\varepsilon$ . One way would be to use a simplex method, and keep track of the  $\varepsilon$ -polynomials, exactly like for Lemke's algorithm. While practical, this would not lead to a polynomial time algorithm. Fortunately, we can again use the fact that  $\varepsilon$  only occurs in polynomials on the right hand side. This allows us to use the same technique as Miltersen and Sørensen [Miltersen and Sørensen 2010, Proposition 1] used for computing a quasi-perfect equilibrium of a zero-sum extensive form game in polynomial time. They show that as long as  $\varepsilon$  only occurs on the right hand side,  $\varepsilon^* = \frac{1}{2}n^{-n-1}V^{-2n-1}$  is sufficiently small to work as an infinitesimal, where  $n$  is the number of constraints and  $V$  is the maximal absolute value of coefficients of the LP. This  $\varepsilon^*$  only requires a polynomial number of bits to represent, so constructing (14) with an explicit  $\varepsilon = \varepsilon^*$  will only cause a polynomial blow-up, and it can thus also be solved in polynomial time. With the computed solution with the explicit  $\varepsilon$  at hand, one can pivot a simplex algorithm to the same solution, while keeping track of the symbolic  $\varepsilon$ . Thus, one can compute a symbolic  $\varepsilon$ -proper equilibrium of a zero-sum game in polynomial time.

## 7. DISCUSSION AND OPEN PROBLEMS

The focus of this paper, both for the zero-sum and for the general-sum case, has been on computing symbolic  $\varepsilon$ -proper equilibria. This has the advantage of allowing easy verification of the properties of the refined equilibrium. This is naturally most important in the general-sum case; if one is only provided with the proper equilibrium of a bimatrix game, finding a symbolic  $\varepsilon$ -proper strategy profile is at least as hard as computing a proper equilibrium from scratch. This can be seen by taking any bimatrix game, and adding one new strategy  $\perp$  to each player. If either player plays  $\perp$ , both players receive some fixed payoff that is higher than all other possible payoffs. If neither player plays  $\perp$ , they receive the payoff from the original game. The constructed game has a unique (proper) equilibrium:  $(\perp, \perp)$ , but finding a symbolic  $\varepsilon$ -proper equilibrium corresponds to computing at least a proper equilibrium of the original game.

As mentioned in Section 6, the proposed algorithm can potentially speed up Drescher's procedure by only solving a single linear program. The linear program will, however, be larger: instead of solving  $O(n)$  linear programs each with  $O(n)$  rows and columns, the presented approach solves one linear program with  $O(n \log n)$  rows and columns. The additional rows and column are, however, sparse, and the total number of non-zero entries is asymptotically the same as for the linear programs of Drescher's procedure.

The presented algorithms works for bimatrix games. An interesting idea would be to try to extend it to succinct representations like the extensive form. This was done for the zero-sum case by Miltersen and Sørensen [Miltersen and Sørensen 2008], but their approach offers no hope of working for general sum games. We therefore phrase the following open problem:

**OPEN PROBLEM 7.1.** *Is finding a strategic-form proper equilibrium of a given two-player extensive-form general-sum game in PPAD?*

While the above discussion shows that it is PPAD-hard to expand a proper equilibrium of a bimatrix game to a symbolic  $\varepsilon$ -proper equilibrium, there might be an easier way to verify whether a given strategy profile is a proper equilibrium. The approach of Belhaiza et.al. does not provide hope here, as their approach includes solving quadratic programs, which are NP-hard to solve. We therefore pose the following open problem:

**OPEN PROBLEM 7.2.** *Given a strategy profile of a bimatrix game, can it be determined in polynomial time whether it is a proper equilibrium?*

## ACKNOWLEDGMENTS

I would like to thank Michel Goemans, Peter Bro Miltersen, Rahul Savani, and Marcin Jurdziński for valuable discussion.

## References

- AJTAI, M., KOMLÓS, J., AND SZEMERÉDI, E. 1983. An  $O(n \log n)$  sorting network. In *Proceedings of the 15th Annual ACM Symposium on the Theory of Computing*. 1–9.
- BELHAIZA, S., AUDET, C., AND HANSEN, P. 2012. On proper refinement of nash equilibria for bimatrix games. *Automatica* 48, 2, 297 – 303.
- BLUME, L., BRANDENBURGER, A., AND DEKEL, E. 1991. Lexicographic probabilities and equilibrium refinements. *Econometrica* 59, 81–98.
- CHEN, X. AND DENG, X. 2006. Settling the complexity of two-player Nash equilibrium. In *Proceedings of 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*. 261–272.
- COTTLE, R. W., PANG, J.-S., AND STONE, R. E. 2009. *The Linear Complementarity Problem*. SIAM.
- DANTZIG, G., ORDEN, A., AND WOLFE, P. 1955. The generalized simplex method for minimizing a linear form under linear inequality restraints. *Pacific Journal of Mathematics* 5, 183–195.
- DASKALAKIS, C., GOLDBERG, P. W., AND PAPADIMITRIOU, C. H. 2006. The complexity of computing a nash equilibrium. In *Proceedings of the 38th Annual ACM Symposium on the Theory of Computing (STOC'06)*. 71–78.
- DRESHER, M. 1961. *The Mathematics of Games of Strategy: Theory and Applications*. Prentice-Hall.
- GOEMANS, M. 2009. Smallest compact formulation for the permutahedron. <http://math.mit.edu/~goemans/PAPERS/permutahedron.pdf>.
- GOEMANS, M. 2010. Personal communication.
- KARMARKAR, N. 1984. A new polynomial-time algorithm for linear programming. *Combinatorica* 4, 4, 373–395.
- KHACHIYAN, L. 1979. A polynomial algorithm in linear programming. *Soviet Mathematics Doklady* 20, 191–194.
- KOHLBERG, E. AND MERTENS, J.-F. 1986. On the strategic stability of equilibria. *Econometrica* 54, 1003–1037.

- KOLLER, D., MEGIDDO, N., AND VON STENGEL, B. 1996. Efficient computation of equilibria for extensive form games. *Games and Economic Behavior* 14, 247–259.
- KREPS, D. M. AND WILSON, R. 1982. Sequential equilibria. *Econometrica* 50, 4, 863–894.
- KUHN, H. 1953. Extensive games and the problem of information. *Annals of Mathematical Studies* 28, 193–216.
- LEMKE, C. 1965. Bimatrix equilibrium points and mathematical programming. *Management Science* 11, 681–689.
- MILTERSEN, P. B. AND SØRENSEN, T. B. 2006a. Computing proper equilibria of zero-sum games. In *Proceedings of the 5th International Conference on Computers and Games*. LNCS Series, vol. 4630. Springer, 200–211.
- MILTERSEN, P. B. AND SØRENSEN, T. B. 2006b. Computing sequential equilibria for two-player games. In *Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithms*. 107–116.
- MILTERSEN, P. B. AND SØRENSEN, T. B. 2008. Fast algorithms for finding proper strategies in game trees. In *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms*.
- MILTERSEN, P. B. AND SØRENSEN, T. B. 2010. Computing a quasi-perfect equilibrium for a two-player game. *Economic Theory* 42, 1, 175–192.
- MYERSON, R. B. 1978. Refinements of the Nash equilibrium concept. *International Journal of Game Theory* 15, 133–154.
- POTTERS, J. A. M. AND TIJS, S. H. 1992. The nucleolus of a matrix game and other nucleoli. *Mathematics of Operations Research* 17, 1, 164–174.
- SCHMEIDLER, D. 1969. The nucleolus of a characteristic function game. *SIAM Journal on Applied Mathematics* 17, 6, pp. 1163–1170.
- SELTEN, R. 1965. Spieltheoretische behandlung eines oligopolmodells mit nachfragerträglichkeit. *Zeitschrift für die gesamte Staatswissenschaft* 12, 301–324.
- SELTEN, R. 1975. A reexamination of the perfectness concept for equilibrium points in extensive games. *International Journal of Game Theory* 4, 25–55.
- TODD, M. J. 1976. Orientation in complementary pivot algorithms. *Mathematics of Operations Research* 1, 1, pp. 54–66.
- VAN DAMME, E. 1984. A relation between perfect equilibria in extensive form games and proper equilibria in normal form games. *International Journal of Game Theory* 13, 1–13.
- VAN DAMME, E. 1991. *Stability and Perfection of Nash Equilibria* 2nd Ed. Springer-Verlag.
- YAMAMOTO, Y. 1993. A path-following procedure to find a proper equilibrium of finite games. *International Journal of Game Theory* 22, 249–259.