

# An Optimization Approach for Approximate Nash Equilibria

Haralampos Tsaknakis and Paul G. Spirakis

**Abstract.** In this paper we propose a new methodology for determining approximate Nash equilibria of noncooperative bimatrix games, and based on that, we provide an efficient algorithm that computes 0.3393-approximate equilibria, the best approximation to date. The methodology is based on the formulation of an appropriate function of pairs of mixed strategies reflecting the maximum deviation of the players' payoffs from the best payoff each player could achieve given the strategy chosen by the other. We then seek to minimize such a function using descent procedures. Because it is unlikely to be able to find global minima in polynomial time, given the recently proven intractability of the problem, we concentrate on the computation of stationary points and prove that they can be approximated arbitrarily closely in polynomial time and that they have the above-mentioned approximation property. Our result provides the best  $\epsilon$  to date for polynomially computable  $\epsilon$ -approximate Nash equilibria of bimatrix games. Furthermore, our methodology for computing approximate Nash equilibria has not been used by others.

## 1. Introduction

Ever since it was proved that the problem of finding exact Nash equilibria is intractable in the sense that it is PPAD-complete even for two-player games [Chen and Deng 06], attention has been focused on finding  $\epsilon$ -approximate such equilibria for  $\epsilon > 0$ . In this respect, simple algorithms have recently been provided for finding approximate equilibria for constant  $\epsilon = \frac{3}{4}$  and  $\epsilon = \frac{1}{2}$  [Daskalakis et al. 06, Kontogiannis et al. 06] for general bimatrix games (and for positively normalized payoff matrices) based on examining small supports of 1 or 2 for

either player. A well-known result provides 0.38-approximate Nash equilibria of normalized bimatrix games in polynomial time [Daskalakis et al. 07]. Concurrently with us, [Boss et al. 07] gave an approach based on [Kontogiannis and Spirakis 07] that provides 0.36-approximate Nash equilibria of normalized bimatrix games. Furthermore, it has been shown [Chen et al. 06] that the more general approximation problem of finding a fully polynomial-time approximation scheme for any  $\epsilon > 0$  has similar complexity with the problem of finding exact Nash equilibria.

For a different, stronger, notion of approximation, i.e., the well-supported approximate Nash equilibria, the best known result so far provides 0.658-approximate well-supported equilibria for normalized bimatrix games in polynomial time [Kontogiannis and Spirakis 07].

Most of the reported investigations of finding approximate equilibria for constant  $\epsilon$  are based on the examination of small supports of the strategy sets of the players, and the algorithms presented are based on brute-force search over all such supports.

In this work we adopt a different approach that does not rely on any pre-specified small supports nor on an indiscriminate search over all small support strategies. We define an equivalent optimization problem in the strategy spaces of both players and attempt to obtain a stationary point of a specific function that measures the maximum deviation of the players' payoffs from the best payoff each player could achieve given the strategy chosen by the other. We do so through a descent procedure along feasible directions in the strategy spaces of both players simultaneously.

Feasible descent directions are computed by solving linear programming problems. Also, by solving similar linear programs we can determine whether there is a descent direction at any given point in the strategy spaces. If a descent direction does not exist, then we have reached a stationary point. We prove that at any stationary point of that function we obtain strategy pairs such that at least one of them is an 0.3393-approximate Nash equilibrium. We also prove that an almost stationary point of the function can be reached in polynomial time with respect to the input data of the game, and that point suffices to get arbitrarily close to 0.3393. Our work can also be accessed as a full technical report (revised) in [Tsaknakis and Spirakis 07].

## 2. Definitions and Notation

Let  $R, C$  denote the  $m \times n$  row and column players' payoff matrices respectively, for  $m, n$  any positive integers. We assume that both payoff matrices are posi-

tively normalized, i.e., all their entries belong to  $[0, 1]$  (without loss of generality any game can be equivalently transformed to a positively normalized game by appropriate shifting and scaling each of the payoff matrices).

Let us denote by  $e_k$  the  $k$ -dimensional column vector having all its entries equal to 1 (for positive integer  $k$ ). Let

$$\Delta_k = \{u : u \in R^k, u \geq 0, e_k^\tau u = 1\}$$

be the  $k$ -dimensional standard simplex (superscript  $\tau$  denotes transpose).

Also, for any vector  $u \in R^k$ , we define the following:

$$\text{supp}(u) = \{i \in (1, k) : u_i \neq 0\}$$

is the support index subset of  $u \in R^k$ , and also

$$\text{suppmax}(u) = \{i \in (1, k) : u_i \geq u_j \forall j \in (1, k)\}$$

is the index subset for which all entries are equal to the maximum entry of  $u \in R^k$ .

We also denote by

$$\max(u) = \{u_i : u_i \geq u_j, \text{ for all } j\}$$

the value of the maximum entry of the vector, and by

$$\max_S(u) = \{u_i, i \in S : u_i \geq u_j, \text{ for all } j \in S\}$$

the value of the maximum entry of the vector within an index subset  $S \subset (1, k)$ .

Finally, we denote by  $\overline{S}$  the complement of an index set  $S$ , i.e.,  $\overline{S} = \{i \in (1, k), i \notin S\}$ .

The problem of finding an  $\epsilon$ -approximate Nash equilibrium in the game  $(R, C)$ , for some  $\epsilon \geq 0$ , is to compute a pair of strategies  $\overline{x}$  in  $\Delta_m$  and  $\overline{y}$  in  $\Delta_n$  such that the following relationships hold:

$$x^\tau R \overline{y} \leq \overline{x}^\tau R \overline{y} + \epsilon \quad \text{for all } x \in \Delta_m$$

and

$$\overline{x}^\tau C y \leq \overline{x}^\tau C \overline{y} + \epsilon \quad \text{for all } y \in \Delta_n.$$

### 3. Optimization Formulation

Key to our approach is the definition of the following continuous function mapping  $\Delta_m \times \Delta_n$  into  $[0, 1]$ :

$$f(x, y) = \max\{\max(Ry) - x^T Ry, \max(C^T x) - x^T Cy\}.$$

It is evident that  $f(x, y) \geq 0$  for all  $(x, y) \in \Delta_m \times \Delta_n$  and that exact Nash equilibria of  $(R, C)$  correspond to pairs of strategies such that  $f(x, y) = 0$ . Furthermore,  $\epsilon$ -approximate equilibria correspond to strategy pairs that satisfy  $f(x, y) \leq \epsilon$ . This function represents the maximum deviation of the players' payoffs from the best payoff each player could achieve given the strategy chosen by the other.

An optimization formulation based on mixed integer programming methods was suggested in [Sandholm et al. 05]. However, no approximation results were obtained there.

The function  $f(x, y)$  is not jointly convex with respect to both  $x$  and  $y$ . However, it is convex in  $x$  alone if  $y$  is kept fixed and vice versa.

Let us define the two ingredients of the function  $f(x, y)$  as follows:  $f_R(x, y) = \max(Ry) - x^T Ry$  and  $f_C(x, y) = \max(C^T x) - x^T Cy$ .

From any point in  $(x, y) \in \Delta_m \times \Delta_n$  we consider variations of  $f(x, y)$  along feasible directions in both players' strategy spaces of the following form:

$$(1 - \epsilon) \begin{bmatrix} x \\ y \end{bmatrix} + \epsilon \begin{bmatrix} x' \\ y' \end{bmatrix},$$

where,  $0 \leq \epsilon \leq 1, (x', y') \in \Delta_m \times \Delta_n$  (the vectors in brackets are  $(m + n)$ -dimensional column vectors). The variation of the function along such a feasible direction is defined by the following relationship:

$$Df(x, y, x', y', \epsilon) = f(x + \epsilon(x' - x), y + \epsilon(y' - y)) - f(x, y).$$

We have derived an explicit formula for  $Df(x, y, x', y', \epsilon)$  (see the appendix, Section 8), which is a piecewise quadratic function of  $\epsilon$ , and the number of switches of the linear terms of the function is at most  $m + n$ . Therefore, for fixed  $(x', y')$  this function can be minimized with respect to  $\epsilon$  in polynomial time. Furthermore, there always exists a positive number, say  $\epsilon^*$ , such that for any  $\epsilon \leq \epsilon^*$  the coefficient of the linear term of this function of  $\epsilon$  coincides with the gradient, as defined below. The number  $\epsilon^*$  generally depends on both  $(x, y)$  and  $(x', y')$ . (See Section 8.3.)

We define the gradient of  $f$  at the point  $(x, y)$  along an arbitrary feasible direction specified by another point  $(x', y')$  as follows:

$$Df(x, y, x', y') = \lim_{\epsilon \rightarrow 0} \frac{1}{\epsilon} Df(x, y, x', y', \epsilon).$$

The gradient  $Df(x, y, x', y')$  of  $f$  at any point  $(x, y) \in \Delta_m \times \Delta_n$  along a feasible direction (determined by another point  $(x', y') \in \Delta_m \times \Delta_n$ ) provides the rate of decrease (or increase) of the function along that direction. For fixed  $(x, y)$ ,  $Df(x, y, x', y')$  is a convex polyhedral function in  $(x', y')$ . In fact, we have derived the explicit form of  $Df(x, y, x', y')$  as the maximum of two linear forms in the  $(x', y')$  space (see the derivations below and in Section 8.1). At any point  $(x, y)$  we wish to minimize the gradient function with respect to  $(x', y')$  to find the steepest possible descent direction, or to determine that no such descent is possible.

Let us define the following index sets:

$$S_R(y) = \text{suppmax}(Ry), \quad S_C(x) = \text{suppmax}(C^T x).$$

By definition,  $S_R(y) \subset (1, m)$  and  $S_C(x) \subset (1, n)$ . From Section 8.1 we get the following: (a) If  $f_R(x, y) = f_C(x, y)$  then

$$Df(x, y, x', y') = \max(T_1(x, y, x', y'), T_2(x, y, x', y')) - f(x, y),$$

where

$$\begin{aligned} m_1(y') &= \max(Ry') \quad \text{over the subset } S_R(y), \\ m_2(x') &= \max(C^T x') \quad \text{over the subset } S_C(x), \\ T_1(x, y, x', y') &= m_1(y') - x^T Ry' - (x')^T Ry + x^T Ry, \\ T_2(x, y, x', y') &= m_2(x') - x^T Cy' - (x')^T Cy + x^T Cy. \end{aligned}$$

(b) If  $f_R(x, y) > f_C(x, y)$ , then  $Df(x, y, x', y') = T_1(x, y, x', y') - f(x, y)$ ; (c) If  $f_R(x, y) < f_C(x, y)$ , then  $Df(x, y, x', y') = T_2(x, y, x', y') - f(x, y)$ . In cases (b) and (c) the functions  $T_1$  and  $T_2$  are as defined in case (a).

The problem of finding  $Df(x, y)$  as the minimum over all  $(x', y') \in \Delta_m \times \Delta_n$  of the function  $Df(x, y, x', y')$  is a linear programming problem.

This problem can be equivalently expressed as the following minimax problem by introducing appropriate dual variables (we derive it for  $(x, y)$  such that  $f_R(x, y) = f_C(x, y)$ , since this is the most interesting case and the cases in which the two terms are different can be reduced to this by solving a linear programming problem, as we shall see below) as follows: Minimize (over  $x', y'$ ) the maximum (over  $w, z, \rho$ ) of the function

$$[\rho w^T, (1 - \rho)z^T] G(x, y) \begin{bmatrix} y' \\ x' \end{bmatrix},$$

where

- (a) the maximum is taken with respect to dual variables  $w, z, \rho$  such that  $w \in \Delta_m, \text{supp}(w) \subset S_R(y)$  and  $z \in \Delta_n, \text{supp}(z) \subset S_C(x)$  and  $\rho \in [0, 1]$ ;
- (b) the minimum is taken with respect to  $(x', y') \in \Delta_m \times \Delta_n$ ;
- (c) the matrix  $G(x, y)$  is the following  $(m + n) \times (m + n)$  matrix:

$$G(x, y) = \begin{bmatrix} R - e_m x^\tau R & -e_m y^\tau R^\tau + e_m e_m^\tau x^\tau R y \\ -e_n x^\tau C + e_n e_n^\tau x^\tau C y & C^\tau - e_n y^\tau C^\tau \end{bmatrix}.$$

The probability vectors  $w$  and  $z$  play the role of price vectors (or penalty vectors) for penalizing deviations from the support sets  $S_R(y)$  and  $S_C(x)$ , and the parameter  $\rho$  plays the role of a tradeoff parameter between the two parts of the function  $f(x, y)$ . In fact,  $w, z$ , and  $\rho$  are not independent variables, but they are taken all together to represent a single  $(m+n)$ -dimensional probability vector on the left-hand side (the maximizing term) of the linear minimax problem.

Solving the above minimax problem, we obtain  $w, z, \rho, x'$ , and  $y'$  that are all functions of the point  $(x, y)$  and take values in their respective domains of definition. Let us denote by  $V(x, y)$  the value of the solution of the minimax problem at the point  $(x, y)$ . The solution of this problem yields a feasible descent direction (as a matter of fact, the steepest feasible descent direction) for the function  $f(x, y)$  if  $Df(x, y) = V(x, y) - f(x, y) < 0$ . Following such a descent direction, we can perform an appropriate line search with respect to the parameter  $\epsilon$  and find a new point that gives a lower value of the function  $f(x, y)$ . Applying repeatedly such a descent procedure, we will eventually reach a point where no further reduction is possible. Such a point is a stationary point that satisfies  $Df(x, y) \geq 0$ .

In the next section we examine the approximation properties of stationary points. In fact, we prove that given any stationary point we can determine pairs of strategies such that at least one of them is a 0.3393-approximate Nash equilibrium.

#### 4. Approximation Properties of Stationary Points

Let us assume that we have a stationary point  $(x^*, y^*)$  of the function  $f(x, y)$ . Then, based on the above analysis and notation, the following relationship should be true:

$$Df(x^*, y^*) = V(x^*, y^*) - f(x^*, y^*) \geq 0.$$

Let  $(w^*, z^*) \in \Delta_m \times \Delta_n$ ,  $\rho^* \in [0, 1]$ , be a solution of the linear minimax problem (with matrix  $G(x^*, y^*)$ ) with respect to the dual variables corresponding to the pair  $(x^*, y^*)$ . Such a solution should satisfy the relations  $\text{supp}(w^*) \subset S_R(y^*)$  and  $\text{supp}(z^*) \subset S_C(x^*)$ .

Let us define the following quantities:

$$\lambda = \min_{y': \text{supp}(y') \subset S_C(x^*)} \{(w^* - x^*)^T R y'\}$$

and

$$\mu = \min_{x': \text{supp}(x') \subset S_R(y^*)} \{x'^T C(z^* - y^*)\}.$$

From the fact that  $R, C$  are positively normalized, it follows that both  $\lambda$  and  $\mu$  are less than or equal to 1.

At any point  $(x^*, y^*)$  these quantities basically define the rates of decrease (or increase) of the function  $f$  along directions of the form  $(1 - \epsilon)(x^*, y^*) + \epsilon(x^*, y')$  and  $(1 - \epsilon)(x^*, y^*) + \epsilon(x', y^*)$ , i.e., the rates of decrease that are obtained when we keep one player's strategy fixed and move the probability mass of the other player into his own maximum support, toward decreasing his own deviation from the maximum payoff he can achieve.

From the stationarity property of the point  $(x^*, y^*)$  it follows that both  $\lambda$  and  $\mu$  are nonnegative. Indeed, in the opposite case there would be a descent direction, which contradicts the stationarity condition.

Let us define a pair of strategies  $(\hat{x}, \hat{y}) \in \Delta_m \times \Delta_n$  as follows:

$$(\hat{x}, \hat{y}) = \begin{cases} (x^*, y^*) & \text{if } f(x^*, y^*) \leq f(\tilde{x}, \tilde{y}), \\ (\tilde{x}, \tilde{y}) & \text{otherwise,} \end{cases}$$

where

$$(\tilde{x}, \tilde{y}) = \begin{cases} \left( \frac{1}{1+\lambda-\mu} w^* + \frac{\lambda-\mu}{1+\lambda-\mu} x^*, z^* \right) & \text{if } \lambda \geq \mu, \\ \left( w^*, \frac{1}{1+\mu-\lambda} z^* + \frac{\mu-\lambda}{1+\mu-\lambda} y^* \right) & \text{if } \lambda < \mu. \end{cases}$$

We now express the main result of this paper in the following theorem.

**Theorem 4.1.** *The pair of strategies  $(\hat{x}, \hat{y})$  defined above is a 0.3393-approximate Nash equilibrium.*

**Proof.** From the definition of  $(\hat{x}, \hat{y})$  we have

$$f(\hat{x}, \hat{y}) \leq \min\{f(x^*, y^*), f(\tilde{x}, \tilde{y})\}. \tag{4.1}$$

Using the stationarity condition for  $(x^*, y^*)$ , we obtain

$$f(x^*, y^*) \leq V(x^*, y^*).$$

But  $V(x^*, y^*)$  is less than or equal to

$$\rho^* E_1 + (1 - \rho^*) E_2,$$

where

$$E_1 = (w^{*\tau} R y' - x^{*\tau} R y' - x'^{\tau} R y^* + x^{*\tau} R y^*)$$

and

$$E_2 = (z^{*\tau} C^{\tau} x' - x^{*\tau} C y' - x'^{\tau} C y^* + x^{*\tau} C y^*),$$

and this holds for all  $(x', y') \in \Delta_m \times \Delta_n$ .

Setting  $x' = x^*$  and  $y' : \text{supp}(y') \subset S_C(x^*)$  in the above inequality, we get

$$f(x^*, y^*) \leq \rho^* \lambda. \tag{4.2}$$

Next, setting  $y' = y^*$  and  $x' : \text{supp}(x') \subset S_R(y^*)$  in the same inequality, we get

$$f(x^*, y^*) \leq (1 - \rho^*) \mu. \tag{4.3}$$

Now using the definition of the strategy pair  $(\tilde{x}, \tilde{y})$  above and exploiting the inequalities

$$\begin{aligned} (w^* - x^*)^T R z^* &\geq \lambda, & \text{since } \text{supp}(z^*) \subset S_C(x^*), \\ w^{*T} C(z^* - y^*) &\geq \mu, & \text{since } \text{supp}(w^*) \subset S_R(y^*), \end{aligned}$$

we obtain (assume  $\lambda \geq \mu$ )

$$\begin{aligned} f_R(\tilde{x}, \tilde{y}) &= \max\{R\tilde{y}\} - \tilde{x}^T R \tilde{y} \\ &= \max\{Rz^*\} - \left( \frac{1}{1 + \lambda - \mu} w^* + \frac{\lambda - \mu}{1 + \lambda - \mu} x^* \right)^T R z^* \\ &= \max\{Rz^*\} - \frac{1}{1 + \lambda - \mu} w^{*T} R z^* - \frac{\lambda - \mu}{1 + \lambda - \mu} x^{*T} R z^* \\ &\leq \max\{Rz^*\} - x^{*T} R z^* - \frac{\lambda}{1 + \lambda - \mu} \leq \frac{1 - \mu}{1 + \lambda - \mu}. \end{aligned}$$



Similarly, setting  $D = C^T$  yields

$$\begin{aligned}
 f_C(\tilde{x}, \tilde{y}) &= \max\{D\tilde{x}\} - \tilde{x}^T C \tilde{y} \\
 &= \max\left\{ \frac{1}{1 + \lambda - \mu} D w^* + \frac{\lambda - \mu}{1 + \lambda - \mu} D x^* \right\} - \frac{1}{1 + \lambda - \mu} w^{*T} C z^* \\
 &\quad - \frac{\lambda - \mu}{1 + \lambda - \mu} x^{*T} C z^* \\
 &\leq \frac{1}{1 + \lambda - \mu} \max\{D w^*\} + \frac{\lambda - \mu}{1 + \lambda - \mu} \max\{D x^*\} - \frac{1}{1 + \lambda - \mu} w^{*T} C z^* \\
 &\quad - \frac{\lambda - \mu}{1 + \lambda - \mu} \max\{D x^*\} \\
 &= \frac{1}{1 + \lambda - \mu} (\max\{D w^*\} - w^{*T} C y^*) - \frac{1}{1 + \lambda - \mu} (w^{*T} C z^* - w^{*T} C y^*) \\
 &\leq \frac{1 - \mu}{1 + \lambda - \mu}.
 \end{aligned}$$

From the above relationships we obtain

$$f(\tilde{x}, \tilde{y}) \leq \frac{1 - \mu}{1 + \lambda - \mu} \quad \text{for } \lambda \geq \mu. \quad (4.4)$$

(A similar inequality can be obtained if  $\lambda < \mu$  and we interchange  $\lambda$  and  $\mu$ .) In all cases, combining inequalities (4.2), (4.3), (4.4) and using the definition of  $(\hat{x}, \hat{y})$  above, we get the following:

$$f(\hat{x}, \hat{y}) \leq \min\left\{ \rho^* \lambda, (1 - \rho^*) \mu, \frac{1 - \min\{\lambda, \mu\}}{1 + \max\{\lambda, \mu\} - \min\{\lambda, \mu\}} \right\}. \quad (4.5)$$

We can prove that the quantity in (4.5) cannot exceed the number 0.3393 for any  $\rho^*, \lambda, \mu \in [0, 1]$ . For the proof see Section 8.2.

This concludes the proof of our main theorem.  $\square$

Using the above analysis, we can consider some special cases of bimatrix games and obtain the following results:

**Corollary 4.2.** *For bimatrix games  $R, C$  such that  $R_{ij} + C_{ij} \leq 1$  for all entries  $(i, j) \in (1, m) \times (1, n)$ , any stationary point yields a  $\frac{1}{4}$ -approximate Nash equilibrium.*

**Proof.** This result follows from

$$\lambda + \mu \leq w^{*T} (R + C) z^* \leq 1$$

and

$$\frac{\lambda\mu}{\lambda + \mu} \leq \frac{\lambda + \mu}{4}, \quad \text{for } \lambda, \mu \geq 0.$$

Notice that the conditions of the above corollary contain the category of win-lose games (whose payoffs are either 0 or 1) that do not have trivial pure Nash equilibria.  $\square$

**Corollary 4.3.** *For constant-sum bimatrix games ( $R_{ij} + C_{ij} = \text{constant}$  for all  $(i, j) \in (1, m) \times (1, n)$ ), a stationary point is an exact Nash equilibrium.*

**Proof.** This result follows from

$$(w^* - x^*)^T (R + C)(z^* - y^*) = 0$$

(which holds because  $w^*, x^*, z^*, y^*$  are probability vectors) and the relationships

$$\begin{aligned} (w^* - x^*)^T R z^* &\geq \lambda, \\ w^{*T} C(z^* - y^*) &\geq \mu, \\ (w^* - x^*)^T R y^* &= x^{*T} C(z^* - y^*) = f(x, y). \end{aligned}$$

Notice that at a stationary point for a constant-sum game we should always have  $\lambda = \mu = 0$ .  $\square$

## 5. Descent Procedure

A stationary point of any general linear complementarity problem can be approximated arbitrarily close in polynomial time via the method presented in [Ye 93]. We give here an alternative approach, directly applicable to our problem.

We present here an algorithm for finding a pair of strategies that achieve the 0.3393 approximation bound. The algorithm is based on a descent procedure of the function  $f(x, y)$ ,  $(x, y) \in \Delta_m \times \Delta_n$ , and consists of the following steps (set  $b = 0.3393$ ):

1. Start with an arbitrary  $(x, y) = (x_0, y_0)$  in  $\Delta_m \times \Delta_n$  (e.g., the uniform distribution). Produce another pair  $(x, y)$  with lower value of  $f(x, y)$  and for which  $f_R(x, y) = f_C(x, y)$  as follows:
  - (a) If  $f_R(x_0, y_0) > f_C(x_0, y_0)$ , keep  $y_0$  fixed and solve the linear programming problem to minimize (over  $x \in \Delta_m$ )

$$\max(Ry_0) - x^T Ry_0$$

under the constraints

$$\max(C^\tau x) - x^\tau C y_0 \leq \max(R y_0) - x^\tau R y_0.$$

- (b) If  $f_R(x_0, y_0) < f_C(x_0, y_0)$ , keep  $x_0$  fixed and solve the linear programming problem to minimize (over  $y \in \Delta_n$ )

$$\max(C^\tau x_0) - x_0^\tau C y$$

under the constraints

$$\max(R y) - x_0^\tau R y \leq \max(C^\tau x_0) - x_0^\tau R y.$$

2. Solve the linear minimax problem with the matrix  $G(x, y)$  as defined in Section 3. Compute the value of  $V(x, y)$ , the pair of strategies  $(x', y')$ , the index sets  $S_R(y) \subset (1, m)$ ,  $S_C(x) \subset (1, n)$ , the vectors  $w, z$ , the parameter  $\rho$ , and the values of  $\lambda, \mu$  as defined in Sections 3 and 4 for the current point  $(x, y)$ . Also determine the pair of strategies  $(\tilde{x}, \tilde{y})$  as defined in Section 4.
3. If at least one of the following conditions is true, stop and exit: a pair of strategies achieving the approximation bound  $b$  has been found.
  - (i)  $V(x, y) - f(x, y) \geq 0$  (stationary condition: either  $f(x, y)$  or  $f(\tilde{x}, \tilde{y})$  is  $\leq b$ );
  - (ii)  $f(x, y) \leq b$ ;
  - (iii)  $f(\tilde{x}, \tilde{y}) \leq b$ ;
  - (iv)  $f(x', y') \leq b$ ;
  - (v)  $f(x', y) \leq b$ ;
  - (vi)  $f(x, y') \leq b$ .
4. If none of the conditions of step 3 is satisfied, compute the minimum with respect to  $\epsilon$  of the function  $f(x + \epsilon(x' - x), y + \epsilon(y' - y))$  along the direction specified by the pair  $(x', y')$  found in step 2, and set  $(x, y) = (x + \epsilon(x' - x), y + \epsilon(y' - y))$  (such a minimization with respect to  $\epsilon$  can be performed in polynomial time, as mentioned earlier, since the number of switches of the linear terms of the piecewise quadratic function cannot exceed  $m + n$ ). Furthermore, if for the new pair  $(x, y)$  we have  $f_R(x, y) \neq f_C(x, y)$ , solve the linear programming problem specified in step 1 and compute the new  $(x, y)$  with lower value of the function  $f(x, y)$  and for which  $f_R(x, y) = f_C(x, y)$ .

Go to Step 2. End of descent.

In regard to the number of steps that are required for convergence and exit, we provide a convergence analysis in Section 8.3 that shows that the algorithm converges in a polynomial number of iterations.

## 6. The Complexity of Our Algorithm

Our algorithm is basically the procedure descent of the function  $f(x, y)$ . As proven in Section 8.3, the number  $q$  of descent steps required for convergence to a  $\delta$ -stationary point, given any  $\delta > 0$ , is less than  $O(\frac{1}{\delta^2})$ , and that suffices to get a  $0.3393 + \delta$ -approximate equilibrium. In fact, the latter complexity bound has recently been improved [Tsaknakis et al. 08] to  $O(\frac{1}{\delta} \log(\frac{1}{\delta}))$ .

So, the total time complexity of our method is  $O(\frac{1}{\delta} \log(\frac{1}{\delta})) T_{LP}(n)$  time (when  $n \geq m$ ), where  $T_{LP}(n)$  is the time to solve a linear program of size  $n$ . Thus, our method is an FPTAS with respect to approximating a stationary point and hence an approximate equilibrium of the stated quality.

An arbitrary point  $(x, y) \in \Delta_m \times \Delta_n$  can be used to initialize the algorithm.

## 7. Discussion and Future Work

It is known from [Bellare and Rogaway 95] that (even in a weaker sense) there is no polynomial-time  $\mu$ -approximation of the optimal value of the problem  $\min\{x^T Qx, | Bx = b, 0 \leq x \leq e\}$  for some  $\mu \in (0, \frac{1}{3})$ , unless  $P = NP$ . Of course, here  $\mu$  is a multiplicative relative accuracy, and the reduction that they use involves matrices that are different from those in our case. However, this gives evidence that going below  $\frac{1}{3}$  in the approximation of equilibria will probably require a radically different approach (if any), perhaps probabilistic. We are currently working on this.

## 8. Appendix: Proofs of Some Results

### 8.1. Derivation of the Gradient Function

Using the definitions for any  $(x, y) \in \Delta_m \times \Delta_n$ , i.e.,

$$\begin{aligned} f_R(x, y) &= \max(Ry) - x^T Ry, \\ f_C(x, y) &= \max(C^T x) - x^T Cy, \\ f(x, y) &= \max\{f_R(x, y), f_C(x, y)\}, \end{aligned}$$

we have, for any  $(x', y') \in \Delta_m \times \Delta_n$  and any  $\epsilon \in [0, 1]$ , that

$$Df(x, y, x', y', \epsilon) = f(x + \epsilon(x' - x), y + \epsilon(y' - y)) - f(x, y).$$

This can be written (analytically) as

$$\begin{aligned} & \max\{f_R(x + \epsilon(x' - x), y + \epsilon(y' - y)), f_C(x + \epsilon(x' - x), y + \epsilon(y' - y))\} \\ & - \max\{f_R(x, y), f_C(x, y)\}, \end{aligned}$$

and this is actually  $\max(K_1, K_2)$ , where

$$K_1 = \epsilon Df_R + \Lambda f_R - \epsilon^2 Hf_R - (1 - \epsilon) \max\{0, f_C(x, y) - f_R(x, y)\}$$

and also

$$K_2 = \epsilon Df_C + \Lambda f_C - \epsilon^2 Hf_C - (1 - \epsilon) \max\{0, f_R(x, y) - f_C(x, y)\},$$

where now the functions  $Df_R, \Lambda f_R, Hf_R, Df_C, \Lambda f_C, Hf_C$  are as defined below:

$$\begin{aligned} Df_R(x, y, x', y') &= \{\max(Ry') \text{ over } S_R(y)\} - x^\tau Ry' - x'^\tau Ry + x^\tau Ry - f(x, y), \\ Hf_R(x, y, x', y') &= (x' - x)^\tau R(y' - y), \\ Df_C(x, y, x', y') &= \{\max(C^\tau x') \text{ over } S_C(x)\} - x^\tau Cy' - x'^\tau Cy + x^\tau Cy - f(x, y), \\ Hf_C(x, y, x', y') &= (x' - x)^\tau C(y' - y). \end{aligned}$$

In order to define  $\Lambda f_R, \Lambda f_C$  we remind the reader that  $S_R(y) = \text{suppmax}(Ry)$  and that  $S_C(x) = \text{suppmax}(C^\tau x)$ , and we will also use their complements, where  $\bar{S}_R(y)$  is the complement of  $S_R(y)$  in the index set  $\{1, m\}$  and  $\bar{S}_C(x)$  is the complement of  $S_C(x)$  in the index set  $\{1, n\}$ .

Now let  $M_y$  be the maximum of  $Ry$  over  $S_R(y)$ ,  $M_{y'}$  the maximum of  $Ry'$  over  $S_R(y)$ ,  $M_x$  the maximum of  $C^\tau x$  over  $S_C(x)$ ,  $M_{x'}$  the maximum of  $C^\tau x'$  over  $S_C(x)$ , and  $\Lambda f_R(x, y, x', y', \epsilon)$  the maximum of 0 and the maximum over  $\bar{S}_R(y)$  of  $(I(y, y') + J(y))$ , where  $I(y, y') = \epsilon((Ry' - e_m M_{y'}) + (M_y e_m - Ry))$  and  $J(y) = -(M_y e_m - Ry)$ .

Also,  $\Lambda f_C(x, y, x', y', \epsilon)$  is also the maximum of 0 and the maximum over  $\bar{S}_C(x)$  of  $(I(x, x') + J(x))$ , where  $I(x, x') = \epsilon((C^\tau x' - e_n M_{x'}) + (M_x e_n - C^\tau x))$  and  $J(x) = -(M_x e_n - C^\tau x)$ .

From the above equations, the gradient at the point  $(x, y) \in \Delta_m \times \Delta_n$  along a feasible direction specified by some  $(x', y') \in \Delta_m \times \Delta_n$  can be determined by letting  $\epsilon$  go to 0, and we obtain, finally,

$$Df(x, y, x', y') = \begin{cases} \max(Df_R, Df_C) & \text{if } f_R(x, y) = f_C(x, y), \\ Df_R & \text{if } f_R(x, y) > f_C(x, y), \\ Df_C & \text{if } f_R(x, y) < f_C(x, y). \end{cases}$$

### 8.2. Proof of the Approximation Bound

We first observe that  $\min\{\rho^*\lambda, (1 - \rho^*)\mu\} \leq \frac{\lambda\mu}{\lambda + \mu}$ . Indeed, if we assume that  $\rho^*\lambda > \frac{\lambda\mu}{\lambda + \mu}$  and  $(1 - \rho^*)\mu > \frac{\lambda\mu}{\lambda + \mu}$  for some  $\rho^*, \lambda, \mu \in [0, 1]$ , we would have  $\rho^* > \frac{\mu}{\lambda + \mu}$  and  $(1 - \rho^*) > \frac{\lambda}{\lambda + \mu}$ , a contradiction. So

$$f(\hat{x}, \hat{y}) \leq \min \left\{ \frac{\lambda\mu}{\lambda + \mu}, \frac{1 - \min\{\lambda, \mu\}}{1 + \max\{\lambda, \mu\} - \min\{\lambda, \mu\}} \right\}.$$

Set  $\mu = \min\{\lambda, \mu\}$ . For  $\mu \leq \frac{1}{2}$  and since  $\mu \leq \lambda$ , we have

$$\frac{\lambda\mu}{\lambda + \mu} \leq \frac{\lambda \min\{1/2, \lambda\}}{\lambda + \min\{1/2, \lambda\}} \leq \frac{1}{3} < 0.3393.$$

Also, for  $\mu \geq \frac{2}{3}$  we have  $1 - \mu \leq \frac{1}{3}$  and  $\frac{1 - \mu}{1 + \lambda - \mu} \leq 1 - \mu \leq \frac{1}{3} < 0.3393$ , since  $\lambda \geq \mu \geq \frac{2}{3}$ .

Consider now cases for which  $\frac{1}{2} < \mu < \frac{2}{3}$ . If  $\frac{1}{2} < \mu \leq \lambda \leq \frac{2}{3}$ , then  $\frac{\lambda\mu}{\lambda + \mu} \leq \frac{\lambda}{2} \leq \frac{1}{3} < 0.3393$ .

For  $\mu, \lambda$  such that  $\frac{1}{2} < \mu < \frac{2}{3} < \lambda$ , let us define  $\xi = \frac{1 - \mu}{\mu}$ . Obviously,  $\frac{1}{2} < \xi < 1$ . Set  $b = 0.3393$ .

Let us assume that there are  $\mu$  and  $\lambda$  satisfying the above relationships and also satisfying

$$\frac{\lambda\mu}{\lambda + \mu} > b \quad \text{and} \quad \frac{1 - \mu}{1 + \lambda - \mu} > b.$$

Expressing these inequalities in terms of  $\xi$  and  $\lambda$ , we get

$$\frac{\xi(1 - b)}{b(1 + \xi)} > \lambda > \frac{b}{1 - b(1 + \xi)}.$$

Since  $b < \frac{1}{2}$ , the above inequality is equivalent to

$$\xi(1 - b)(1 - b(1 + \xi)) - b^2(1 + \xi) > 0 \Leftrightarrow -\xi^2 b(1 - b) + \xi(1 - 2b) - b^2 > 0.$$

It can be verified by direct calculation that the discriminant of the above quadratic is zero for  $b = 0.3393$ , and the inequality becomes

$$-b(1 - b) \left( \xi - \frac{1 - 2b}{2b(1 - b)} \right)^2 > 0,$$

a contradiction.

In fact, the constant  $b$  is the smallest real solution of the equation

$$4b(1 - b)(1 + b^2) = 1.$$

The bound is attained at  $\mu = 0.582523$  and  $\lambda = 0.81281$ .

### 8.3. Proof of Polynomial Convergence of the Descent Algorithm

Let  $\delta > 0$ . Let  $(x, y)$  be the current pair of strategies obtained during the descent procedure for which  $f_R(x, y) = f_C(x, y)$ . Define the following index sets:

$$S_R(y, \delta) = \{i \in (1, m) : (Ry)_i \geq \max(Ry) - \delta\},$$

$$S_C(x, \delta) = \{j \in (1, n) : (C^T x)_j \geq \max(C^T x) - \delta\}.$$

Denoting by  $\overline{S}_R(y, \delta)$  and  $\overline{S}_C(x, \delta)$  the complements of the above sets in the index sets  $(1, m)$  and  $(1, n)$  respectively, it is evident that  $\max(Ry) - (Ry)_i > \delta$  for all  $i \in \overline{S}_R(y, \delta)$ , and  $\max(C^T x) - (C^T x)_j > \delta$  for all  $j \in \overline{S}_C(x, \delta)$ .

Now let  $(x', y')$  be a descent direction, i.e., a pair of strategies that is a solution of the linear minimax problem as defined in Section 3, whose computation is now based on the larger index sets  $S_R(y, \delta)$  and  $S_C(x, \delta)$  (as defined above) instead of  $S_R(y)$  and  $S_C(x)$ . We also consider all quantities defined in Sections 3 and 8.1 based on the index sets  $S_R(y, \delta)$  and  $S_C(x, \delta)$  (we use the same notation for all of them for simplicity).

Considering the variation of the function  $f(x, y)$  along the direction  $(x', y')$ , we obtain the following relationship for any  $\epsilon \in [0, 1]$ :

$$f(x + \epsilon(x' - x), y + \epsilon(y' - y)) - f(x, y) \leq \max\{\epsilon(V_R(x, y) - f(x, y)) + \Lambda f_R - \epsilon^2 H f_R, \epsilon(V_C(x, y) - f(x, y)) + \Lambda f_C - \epsilon^2 H f_C\},$$

where  $H f_R, H f_C, \Lambda f_R, \Lambda f_C$  are as defined in Section 8.1, and

$$V_R(x, y) = \{\max(Ry') \text{ over } S_R(y, \delta)\} - x^T R y' - x'^T R y + x^T R y,$$

$$V_C(x, y) = \{\max(C^T x') \text{ over } S_C(x, \delta)\} - x^T C y' - x'^T C y + x^T C y.$$

Let

$$V(x, y) = \max\{V_R(x, y), V_C(x, y)\}.$$

The quantities  $\Lambda f_R, \Lambda f_C$  are both piecewise linear convex functions of  $\epsilon$  and are equal to 0 for  $\epsilon \leq \epsilon^*$ , where  $\epsilon^*$  is given by  $\epsilon^* = \min\{\epsilon_1^*, \epsilon_2^*, 1\}$ , and  $\epsilon_1^*$  is the minimum over  $i \in \overline{S}_R(y, \delta)$  of

$$\frac{\max(Ry) - (Ry)_i}{\max(Ry) - (Ry)_i + (Ry')_i - \max_{S_R(y)}(Ry')}$$

for some  $i \in \overline{S}_R(y, \delta)$ ,

and  $\epsilon_2^*$  is the minimum over  $j \in \overline{S}_C(x, \delta)$  of

$$\frac{\max(C^T x) - (C^T x)_j}{\max(C^T x) - (C^T x)_j + (C^T x')_j - \max_{S_C(x)}(C^T x')}$$

for some  $j \in \overline{S}_C(x, \delta)$ .

Since  $\max(Ry) - (Ry)_i > \delta$  for all  $i \in \overline{S}_R(y, \delta)$  and  $\max(C^\tau x) - (C^\tau x)_j > \delta$  for all  $j \in \overline{S}_C(x, \delta)$ , we must have

$$\epsilon^* > \frac{\delta}{1 + \delta}.$$

The quantities  $Hf_R, Hf_C$  appearing in the quadratic terms of  $\epsilon$  are bounded from below by  $V_R(x, y) - 1$  and  $V_C(x, y) - 1$  respectively. Defining the new value of  $f$  as  $f_{\text{new}}$  and dropping the arguments (for simplicity), we deduce that the minimum possible descent that can be achieved for the function  $f(x, y)$  can be computed from the following relationship:

$$f_{\text{new}} - f \leq \epsilon(V - f) + \epsilon^2(1 - V),$$

which holds for every  $\epsilon$  such that  $0 \leq \epsilon \leq \frac{\delta}{1 + \delta}$ . From the last relationship, setting  $\epsilon = \frac{\delta}{1 + \delta}$ , we obtain the following:

(a) If  $V - f \leq -\delta$ , then

$$f_{\text{new}} \leq \left(1 - \left(\frac{\delta}{1 + \delta}\right)^2\right) f.$$

(b) If  $V - f > -\delta$ , then we have a  $\delta$ -stationary point.

Since at every step of the algorithm we either have a reduction of the function  $f$  as in (a) above or a  $\delta$ -stationary point as in (b), it turns out that the algorithm terminates with a  $\delta$ -stationary point in  $O\left(\frac{1}{\delta^2}\right)$  steps, and this holds for any  $\delta > 0$ .

Next, we consider the implications of  $\delta$ -stationarity on the approximation bound  $b = 0.3393$ . Following similar arguments, definitions, and notation as in Section 4 and in the proof of Theorem 4.1 (using now the index sets  $S_R(y, \delta)$  and  $S_C(x, \delta)$ ), we obtain the following relationships at a  $\delta$ -stationary point  $(x, y)$ :

$$\begin{aligned} f(x, y) &\leq \rho\lambda + (1 - \rho)\delta, \\ f(x, y) &\leq (1 - \rho)\mu + \rho\delta. \end{aligned}$$

(Notice that the quantities  $\lambda, \mu, \rho$  depend in general on  $\delta$ .)

From the previous relationships, it can be easily verified that  $\delta$ -stationarity yields a  $0.3393 + \delta$ -approximate Nash equilibrium.

**Acknowledgments.** This work has been partially supported by the IST 6th Framework Programme of the European Union under contract 001907 DELIS.



## References

- [Bellare and Rogaway 95] M. Bellare and P. Rogaway. “The Complexity of Approximating a Nonlinear Program.” *Mathematical Programming* 69 (1995), 429–441.
- [Boss et al. 07] H. Boss, J. Byrka, and E. Markakis. “New Algorithms for Approximate Nash Equilibria in Bimatrix Games.” Paper presented at the Third International Workshop on Internet and Network Economics (WINE 2007), San Diego, CA, December 12–14, 2007.
- [Chen and Deng 06] X. Chen and X. Deng. “Settling the Complexity of 2-Player Nash Equilibrium.” In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, pp. 261–272. Los Alamitos, CA: IEEE Press, 2006.
- [Chen et al. 06] X. Chen, X. Deng, and S. Teng. “Computing Nash Equilibria: Approximation and Smoothed Complexity.” In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, pp. 603–612. Los Alamitos, CA: IEEE Press, 2006.
- [Daskalakis et al. 06] C. Daskalakis, A. Mehta, and C. Papadimitriou. “A Note on Approximate Nash Equilibria.” In *Internet and Network Economics Second International Workshop, WINE 2006, Patras, Greece, December 15–17, 2006, Proceedings*, Lecture Notes in Computer Science 4286, pp. 297–306. New York: Springer, 2006.
- [Daskalakis et al. 07] C. Daskalakis, A. Mehta, and C. Papadimitriou. “Progress in Approximate Nash Equilibria.” In *Proceedings of the 8th ACM Conference on Electronic Commerce*, pp. 355–358. New York: ACM Press, 2007.
- [Kontogiannis et al. 06] S. Kontogiannis, P. Panagopoulou, and P. G. Spirakis. “Polynomial Algorithms for Approximating Nash Equilibria in Bimatrix Games.” In *Internet and Network Economics Second International Workshop, WINE 2006, Patras, Greece, December 15–17, 2006, Proceedings*, Lecture Notes in Computer Science 4286, pp. 282–296. New York: Springer, 2006.
- [Kontogiannis and Spirakis 07] S. Kontogiannis and P. G. Spirakis. “Efficient Algorithms for Constant Well Supported Approximate Equilibria of Bimatrix Games.” In *Automata, Languages and Programming: 34th International Colloquium, ICALP 2007, Wroclaw, Poland, July 9–13, 2007, Proceedings*, Lecture Notes in Computer Science 4596, pp. 595–606. Berlin: Springer, 2007.
- [Sandholm et al. 05] T. Sandholm, A. Gilpin, and V. Conitzer. “Mixed-Integer Programming Methods for Finding Nash Equilibria.” In *Proceedings of the 20th National Conference on Artificial Intelligence*, Vol. 2, pp. 495–501. Menlo Park, CA: AAAI Press, 2005.
- [Tsaknakis and Spirakis 07] H. Tsaknakis and P.G. Spirakis. “An Optimization Approach for Approximate Nash Equilibria.” Technical report number TR07-067, Electronic Colloquium on Computational Complexity, August 23, 2007.
- [Tsaknakis et al. 08] H. Tsaknakis, P. G. Spirakis, and D. Kanoulas. “Performance Evaluation of a Descent Algorithm for Bi-matrix Games.” In *Internet and Network Economics: 4th International Workshop, WINE 2008, Shanghai, China, December 17–20, 2008, Proceedings*, edited by C. Papadimitriou and S. Zhang, Lecture Notes in Computer Science 5385, pp. 222–230. Berlin: Springer, 2008.

- [Ye 93] Yinyu Ye. “A Fully Polynomial Time Approximation Algorithm for Computing a Stationary Point of the General Linear Complementarity Problem.” *Mathematics of Operations Research* 18:2 (1993), 334–345.

---

Haralampos Tsaknakis, Research Academic Computer Technology Institute (RACTI),  
26500 Rion, Greece (tsaknak@cti.gr)

Paul G. Spirakis, Department of Computer Engineering and Informatics, Patras Uni-  
versity, GR 265 00, Patras, Greece (spirakis@cti.gr)

Received March 20, 2008; accepted January 27, 2009.