On the computational complexity of Nash equilibria for (0, 1) bimatrix games

Bruno Codenotti [a,*,1], Daniel Štefankovič [b,c]

[a] *Toyota Technological Institute at Chicago, Chicago, IL 60637, USA*
[b] *Department of Computer Science, The University of Chicago, Chicago, IL 60637, USA*
[c] *Department of Computer Science, Comenius University, Bratislava, Slovakia*

Abstract

The computational complexity of finding a Nash equilibrium in a nonzero sum bimatrix game is an important open question. We put forward the notion of $(0, 1)$-bimatrix games, and show that some associated computational problems are as hard as in the general case.

Keywords: NP-completeness; Computational complexity; Bimatrix games; Nash equilibrium; Imitation games

1. Introduction

With the advent of the Internet, algorithms and protocols are starting to embed features imported from Game Theory. This has led to a growing interest toward the computational complexity of the fundamental game theoretic notions. In the setting of noncooperative games, particular attention has been given to the computation of Nash equilibria for nonzero sum games, which is considered one of the most important open questions in computational complexity today [7, 8]. Despite an impressive amount of work (see, e.g., [3, 10]) it is still unknown if a Nash equilibrium for these games can be computed in polynomial time, even in the two player case. On the other hand, NP-hardness results are known for the computation of Nash equilibria with additional properties, e.g., with payoffs above a given threshold [1,2].

In this paper we start exploring some complexity questions related to games where the payoff to the players is either zero or one. More precisely, we look at the computation of Nash equilibria for a class of bimatrix games, which we call *simple bimatrix games*

* Corresponding author.
E-mail addresses: bcodenotti@tti-c.org (B. Codenotti), stefanko@cs.uchicago.edu (D. Štefankovič).
[1] On leave from IIT-CNR, Pisa, Italy.

(SBGs from now on), where the payoff matrices are $(0, 1)$ matrices.

By reduction from 3SAT, we show that it is NP-complete to decide whether there is more than one Nash equilibrium in an SBG. The proof of this result also leads to the NP-hardness of finding a Nash equilibrium with payoff at least k for one of the players.

To prove our results, we associate to an SBG a directed graph and we introduce a graph property, which we call *good assignment*. We reduce 3SAT to the existence problem for good assignments (other than a trivial one).

Adopting the terminology from [5], let us call *imitation* SBGs the SBGs where the row player, called the *imitator*, gets payoff 1 if she makes the same move as the opponent, and 0 otherwise.

We show the equivalence between SBGs and imitation SBGs, and prove that there is a one-to-one correspondence between good assignments and Nash equilibrium strategies for the imitator in imitation SBGs.

Our reduction has the following interpretation. In an imitation SBG, there is always one Nash equilibrium corresponding to a win for the imitator, while the existence of another Nash equilibrium, more favorable to the other player, is subject to the satisfiability of a given formula.

Our results can be summarized by the following theorem.

Theorem 1. *It is* NP*-complete*

(a) *to decide whether an SBG has more than one Nash equilibrium*;
(b) *to decide whether an imitation SBG has a Nash equilibrium with nonzero payoff for imitator's opponent.*

2. Background on bimatrix games

We consider SBGs in *strategic* or *normal form*. These games are described in terms of two $(0, 1)$ matrices, containing the *payoffs* of the two players. The rows (resp. columns) of both matrices are indexed by the row (resp. column) player's *pure strategies*.

A mixed strategy consists of a set of pure strategies and a probability distribution (a collection of nonnegative weights adding up to one) which indicates how likely it is that each pure strategy is played. In other words, each player associates to her ith pure strategy a number p_i between 0 and 1, such that $\sum_i p_i = 1$.

Let us consider a two-player game, where each player has n pure strategies, and let x be a mixed strategy of the row player, and y a mixed strategy of the column player. Strategy x is the n-tuple $x = (x_1, x_2, \ldots, x_n)$, where $x_i \geqslant 0$, and $\sum_{i=1}^{n} x_i = 1$. Similarly, $y = (y_1, y_2, \ldots, y_n)$, where $y_i \geqslant 0$, and $\sum_{i=1}^{n} y_i = 1$. Let now $A = (a_{ij})$ be the payoff matrix of the row player. The entry a_{ij} is the payoff to the row player, when she plays her ith pure strategy and the opponent plays the pure strategy j. According to the mixed strategies x and y, the entry a_{ij} contributes to the expected payoff of the row player with weight $x_i y_j$. The expected payoff of the row player can be evaluated by adding up all the entries of A weighted by the corresponding entries of x and y, i.e., the payoff is $\sum_{ij} x_i y_j a_{ij}$. This can be rewritten as $\sum_i x_i \sum_j a_{ij} y_j$, which can be expressed in matrix terms as[2] $x^{\mathrm{T}} A y$. Similarly, the expected payoff of the column player is $x^{\mathrm{T}} B y$.

A pair (x, y) is in Nash equilibrium if $x^{\mathrm{T}} A y \geqslant x'^{\mathrm{T}} A y$, and $x^{\mathrm{T}} B y \geqslant x^{\mathrm{T}} B y'$, for all stochastic n-vectors x' and y'. If the pair (x, y) is in Nash equilibrium, we say that x (resp. y) is a *Nash equilibrium strategy* for the row (resp. column) player. It is well known that a Nash equilibrium in mixed strategies always exists [6].

To avoid trivial pure strategy Nash equilibria, we assume that the matrices A and B do not have entries equal to 1 in the same position. In other words the game does not have outcomes where both players win. On the other hand, there are outcomes where both players lose, because of the nonconstant sum assumption.

3. Hardness results

Let G be a directed graph. Let x be an assignment of nonnegative weights to the vertices of G. We will assume that x is normalized, i.e., $\|x\|_1 = \sum_i x_i = 1$. The *income* $i_x(v)$ of a vertex v is the sum of weights of vertices u which point to v, i.e., $i_x(v) =$

[2] We use the notation x^{T} to denote the transpose of vector x.

$\sum_{u:(u,v)\in G} x_u$. A vertex v is *happy* if it has highest income (i.e., $i_x(v) \geqslant i_x(u)$ for all $u \in G$). A vertex v is *working* if it has nonzero weight (i.e., $x(v) > 0$). An assignment x is *good* if all the working vertices are happy. As we will see later there always exists a good assignment.

Lemma 2. *It is* NP*-complete to decide if there are at least two good assignments in a given graph G.*

Proof. We will show a reduction from 3SAT. Let F be a formula with m clauses and n variables. Without loss of generality we can assume that $m = 3^k$, for some integer $k > 1$. Let graph H have literal vertices $x_i, \bar{x}_i,\ 1 \leqslant i \leqslant n$, clause vertex C and clause filling vertices $v_{1,C}, v_{2,C}$ for each clause C in F. Connect each literal vertex ℓ to clause vertex C, if ℓ is in C. Connect clause filling vertices $v_{1,C}, v_{2,C}$ to the clause vertex C. Add a ternary tree T with edges directed towards the root, such that the clause vertices are the leaves of T. Let a be the root of T. Connect a to the clause filling vertices. This defines H. Now we construct the graph G by adding vertices to H. For each triple d of vertices in $H-a$ we add an equality checking vertex w_d and connect the vertices in d to w_d. We also add vertices $y_i, \bar{y}_i, z_i,\ 1 \leqslant i \leqslant n$, and connect y_i to both x_i and z_i, $\bar{y}_i$ to both $\bar{x}_i$ and z_i, and a to both y_i and $\bar{y}_i$, for $1 \leqslant i \leqslant n$. Finally we add a vertex r and connect all the equality checking vertices, vertex a, and $z_i,\ 1 \leqslant i \leqslant n$, to r. A sketch of this construction is shown in Fig. 1. Clearly the weight assignment which gives r weight 1 is good (nobody earns anything).

We now prove that there is another good weight assignment in G if and only if F is satisfiable.

Assume that F is satisfiable. Fix a satisfying assignment s of F. Assign weight 1 to the satisfied literals and weight 3 to their predecessors (a subset of the y_i and $\bar{y}_i$). (Note that we are using integer weights; we can then derive a normalized assignment by properly scaling all the weights.) Assign weight 1 to each clause vertex and some of its filling vertices so that the income of each clause vertex is 3. Further assign weight 3 to the vertex a and weight 1 to the rest of vertices in T. Assign weight 0 to all the remaining vertices of G. Clearly the obtained assignment is good, as the reader can verify

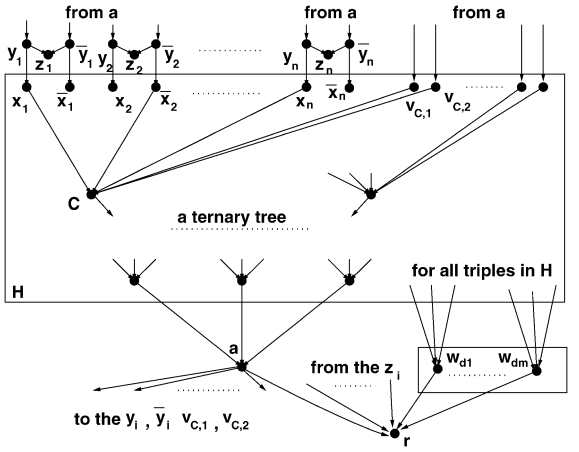

Fig. 1. A sketch of graph G. The large rectangle includes graph H, except for vertex a.

by direct inspection of the status of each type of vertices.

To show the other direction, assume that x is a good assignment in G. If a does not work then the assignment x must give r weight 1 and weight 0 to every other vertex, because $G-a$ is an acyclic graph and r is its unique sink. Hence we can assume that a works. Similarly at least one of the successors of a (other than r) must work, because otherwise a could not have a positive income. Since r cannot have a higher income than a, the weight of the z_i's and of the equality checking vertices is zero. Without loss of generality let the weight of a (and hence also the income of every happy vertex) be 3.

The sum of weights of the predecessors of a must be 3, and hence the weight of any vertex in $H-a$ is at most 1. Assume this is not the case, i.e., that there is a vertex w in $H-a$ with weight larger than 1. Let W' be the set of predecessors of a, if w is not a predecessor of a, and the set of predecessors of w, otherwise. Let W'' be the set containing the two vertices from W' of highest weight. Since the sum of the weights of the vertices in W' is at least 3, the sum of the weights of the vertices in W'' is at least 2, so that the two vertices in W'' together with w have weight strictly larger than 3. This is a contradiction, since the corresponding equality checking vertex has income strictly larger than 3.

Therefore all the vertices in $T-a$ must have weight 1. In particular each clause vertex must be working, and hence be happy. Therefore for each

clause vertex C at least one of its literals must be working, otherwise C would earn at most 2. Note that x_i and $\bar{x}_i$ cannot both be working for otherwise y_i and $\bar{y}_i$ would have weight 3, and hence z_i would earn 6, thus making a unhappy. It follows that the set of working literal vertices induces a satisfying assignment for F. □

We now show the connection between good assignments and Nash equilibria.

Lemma 3. *Let (I, C) be an SBG. Let $G[C]$ be the oriented graph with adjacency matrix C. The Nash equilibrium strategies of the imitator in (I, C) are in one-to-one correspondence with the good assignments in $G[C]$.*

Proof. Let x be a good assignment for $G[C]$. Then the vector of incomes $x^{\mathrm{T}}C$ is maximal on coordinates where x is nonzero. Let y be uniform on entries on which x is nonzero. The vector of pure strategy payoffs for the imitator in (I, C) is $Iy = y$, and hence (x, y) is a Nash equilibrium for (I, C).

To see the other direction, let us consider any Nash equilibrium (x, y) for (I, C). Assume x is not a good assignment for $G[C]$. Then there is a nonzero entry of x, say x_i, such that $(x^{\mathrm{T}}C)_i < (x^{\mathrm{T}}C)_j$, for some j. Therefore $y_i = 0$, which in turn implies that $x_i = 0$, which is a contradiction. □

The following example illustrates the one-to-one correspondence stated in Lemma 3.

Example 4. Let us consider the matrix

$$C = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

It is easy to check that the following are good assignments for $G[C]$:

$$x_1 = \left(\tfrac{1}{2}, \tfrac{1}{4}, \tfrac{1}{4}, 0\right), \qquad x_2 = \left(\tfrac{1}{2}, \tfrac{1}{2}, 0, 0\right),$$
$$x_3 = \left(\tfrac{1}{2}, 0, \tfrac{1}{2}, 0\right), \qquad x_4 = (0, 0, 0, 1).$$

Moreover, let us consider the vectors

$$y_1 = \left(\tfrac{1}{3}, \tfrac{1}{3}, \tfrac{1}{3}, 0\right), \qquad y_2 = \left(\tfrac{1}{2}, \tfrac{1}{2}, 0, 0\right),$$
$$y_3 = \left(\tfrac{1}{2}, 0, \tfrac{1}{2}, 0\right), \qquad y_4 = (0, 0, 0, 1).$$

The pairs (x_i, y_i), for $i = 1, 2, 3, 4$, are Nash equilibria for the game (I, C).

We are now ready to prove the theorem stated in the Introduction.

Proof of Theorem 1. (a) The proof follows from Lemma 2 and from the correspondence between Nash equilibria and good assignments stated in Lemma 3.

(b) The problem of deciding whether an imitation SBG has a Nash equilibrium with payoff at least k for the column player is clearly in NP. The Nash equilibrium corresponding to the good assignment in Lemma 2 in which only r works has payoff zero for the column player, since r is a sink, and the respective row of the adjacency matrix is zero. The Nash equilibria corresponding to good assignments arising from satisfying assignments of F have nonzero payoff for the column player. □

We finally show a general relation between imitation games and bimatrix games. The following lemma implies that finding Nash equilibria of imitation games is not easier than finding Nash equilibria of general games, in a sense to be made precise below.

Lemma 5. *Let A, B be two $m \times n$ matrices with nonnegative entries, where A^{T} and B have no zero row. Let $C = \left(\begin{smallmatrix} 0 & B \\ A^{\mathrm{T}} & 0 \end{smallmatrix}\right)$ and let I be the $(m + n) \times (m + n)$ identity matrix. The Nash equilibria of the game (A, B) are in one-to-one correspondence with the Nash equilibrium strategies of the row player in the game (I, C).*

Proof. Let x, y be a Nash equilibrium of the game (A, B). Let $x'^{\mathrm{T}} = (\alpha x^{\mathrm{T}}, \beta y^{\mathrm{T}})$, where $\alpha, \beta > 0$ are such that $\beta \max_i (Ay)_i = \alpha \max_i (x^{\mathrm{T}}B)_i$ and $\|x'\|_1 = 1$. Note that such α, β exist since $\max_i (Ay)_i > 0$ and $\max_i (x^{\mathrm{T}}B)_i > 0$.

Let y' be uniform on the coordinates on which x' is nonzero. Note that the vector $x'^{\mathrm{T}}C$ of pure strategy payoffs to the imitator's opponent is $(\beta y^{\mathrm{T}}A^{\mathrm{T}}, \alpha x^{\mathrm{T}}B)$, which is maximal on the coordinates played by the imitator's opponent, because x, y is a Nash equilibrium of the game (A, B), and by the definition of α and β. Clearly x', y' is a Nash equilibrium of the game (I, C).

Let x', y' be a Nash equilibrium of the game (I, C). Let $x'^{\mathrm{T}} = (x^{\mathrm{T}}, y^{\mathrm{T}})$. Because of the assumption on A and B, both x and y are nonzero.

Let α, β be such that $\|\alpha x\|_1 = \|\beta y\|_1 = 1$. Then $(\alpha x, \beta y)$ is a Nash equilibrium of the game (A, B). □

Remark 6. In Lemma 5 we have assumed that A^{T} and B have at least one nonzero entry in each row. Note that the adjacency matrix of Lemma 2 does not satisfy this assumption, since it contains a zero row, corresponding to vertex r in the construction of graph G (see Fig. 1).

However we can modify any given bimatrix game (A, B) without changing its equilibrium structure, by adding a constant to any column of A and/or to any row of B. In particular, we can replace any all-zero column of A by a column of all ones, and we can replace any all-zero row of B by a row of all ones.

This transformation allows us to conclude that imitation SBGs are as hard as general SBGs, in the following sense: if we can efficiently decide if there is more than one Nash equilibrium for imitation SBGs, then we can do it for general SBGs (see part (a) of Theorem 1).

However the transformation changes the payoff structure, so that we cannot reach a similar conclusion for part (b) of Theorem 1.

The following example illustrates Lemma 5 and its proof.

Example 7. Let us consider the bimatrix game (I, C), where

$$C = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}.$$

A Nash equilibrium strategy for the imitator is given by $x' = (\frac{1}{2}, 0, 0, \frac{1}{2})$, while a Nash equilibrium strategy for the imitator's opponent is a mixed strategy which is nonzero and uniform on the coordinates where x' is nonzero.

Now consider the bimatrix game (A, B), where $A = \left(\begin{smallmatrix} 1 & 1 \\ 1 & 0 \end{smallmatrix}\right)$ and $B = \left(\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right)$.

Now, from the second part of the proof of Lemma 5, pick $\alpha = \beta = 2$. We readily obtain $x = (1, 0)$, and $y = (0, 1)$, which form a Nash equilibrium for (A, B).

4. Open questions and further work

Despite a lot of effort over the last years, the answer to the fundamental complexity questions in Game Theory has so far remained elusive. SBGs provide a simpler and somewhat more structured framework in which some of these questions still make sense, and might become easier.

Our work on SBGs leaves a number of unanswered questions.

Are SBGs as *hard* as more general bimatrix games? For instance, are they any easier than games where the payoffs can be 0, 1, or 2? Or, rather, is there a polynomial time computable reduction mapping the latter games into SBGs?

The most popular algorithm for computing Nash equilibria for bimatrix games is Lemke–Howson algorithm [3]. There are simple instances of bimatrix games where Lemke–Howson algorithm takes exponential time [9]. Are there lower bounds on the performance of Lemke–Howson algorithm for SBGs?

Quasi polynomial time algorithms are known for the computation of an *approximate Nash equilibrium* for bimatrix games [4]. Is it easier (perhaps polynomial-time) to find an approximate Nash equilibrium for SBGs?

Acknowledgements

We wish to thank two anonymous referees for critical comments that helped to improve the paper. In particular, we are indebted to one of the referees who discovered some gaps in our proofs and suggested how to fix them.

References

[1] V. Conitzer, T. Sandholm, Complexity results about Nash equilibria, in: Internat. Joint Conf. on Artificial Intelligence, 2003.

[2] I. Gilboa, E. Zemel, Nash and correlated equilibria: Some complexity considerations, Games Econom. Behav. 1 (1989) 80–93.

[3] C.E. Lemke, J.T. Howson, Equilibrium points in bimatrix games, J. Soc. Industr. Appl. Math. 12 (1964) 413–423.

[4] R.J. Lipton, E. Markakis, A. Mehta, Playing large games using simple strategies, in: ACM Conf. Electronic Commerce, 2003, pp. 36–41.

[5] A. McLennan, R. Tourky, From Lemke–Howson to Kakutani, Working Paper, Department of Economics, University of Minnesota, 2004.
[6] J. Nash, Non-cooperative games, Ann. of Math. 54 (2) (1951) 286–295.
[7] C.H. Papadimitriou, On the complexity of the parity argument and other inefficient proofs of existence, J. Comput. System Sci. 48 (1994) 498–532.
[8] C.H. Papadimitriou, Algorithms, games, and the Internet, in: STOC 2001.
[9] R. Savani, B. von Stengel, Exponentially many steps for finding a Nash equilibrium in a bimatrix game, CDAM Research Report LSE-CDAM-2004-03, 2004.
[10] B. von Stengel, Computing equilibria for two-person games, in: R.J. Aumann, S. Hart (Eds.), Handbook of Game Theory, vol. 3, North-Holland, Amsterdam, 2002, pp. 1723–1759, Chapter 45.