Electronic Colloquium on Computational Complexity, Report No. 31 (2005)

Pure Nash equilibria in games with a large number of actions*

Carme Àlvarez Joaquim Gabarró Maria Serna

{alvarez,gabarro,mjserna}@lsi.upc.edu
Universitat Politècnica de Catalunya,
ALBCOM Research group
Edifici Ω, Campus Nord
Jordi Girona, 1-3, Barcelona 08034, Spain.

Abstract. We study the computational complexity of deciding the existence of a Pure Nash Equilibrium in multi-player strategic games. We address two fundamental questions: how can we represent a game?, and how can we represent a game with polynomial pay-off functions? Our results show that the computational complexity of deciding the existence of a pure Nash equilibrium in an strategic game depends on two parameters: the number of players and the size of the sets of strategies. In particular we show that deciding the existence of a Nash equilibrium in an strategic game is NP-complete when the number of players is large and the number of strategies for each player is constant, while the problem is Σ_2^p-complete when the number of players is a constant and the size of the sets of strategies is exponential (with respect to the length of the strategies).

Keywords. *Strategic games, Nash equilibria, complexity classes.*

1 Introduction

In recent times a lot of attention has been devoted to the computational aspects of fundamental concepts in game theory like Nash equilibria [16, 3, 4, 11, 13]. However, classic books on game theory [14] do not explore any connection with computability results.

The question that motivates the present work is *which is the complexity of deciding whether a game has a pure Nash equilibrium?* This fundamental question posed by Papadimitriou [16] has initiated a line of research towards understanding the complexity of computing a pure or a mixed Nash equilibrium, see [7, 8, 6, 10, 5, 9]. However in any of those references there is a lack of a clarity and uniformity in the representation of games. Therefore there is a need for a framework that could be used as a basis for analyzing the computational complexity of problems on games.

The main elements that form part of a game are the players and, for each player, their actions and pay-off functions. We note that, for any problem on games to be computationally meaningful, the number of players or the number of actions of each player or both should be large, furthermore the set of actions and the payoff functions could be given in some implicit way. For any of those elements we can consider explicit descriptions, by means of listing the set of actions and tabulating the pay-off functions, what we call the *standard form*, or more succinct representations in which the pay-off functions are described in terms of Turing machines. When considering a Turing machine as part of a description, an additional element is needed in it, the time allowed to the machine. In this way we obtain succinct descriptions of games that are *non-uniformly* described from Turing machines. We can further describe the actions explicitly, by giving the list of the actions allowed to each player, what we call the *general form*, or succinctly, by giving the length of the actions, what we call the *implicit form*.

Another common notion that appears in the literature is that of a *game with polynomial time computable utilities* [10, 6]. A motivating example is the one of *Congestion games* proposed in [6]

* Work partially supported by IST program of the EU under contract IST-2001-33116 (Flags) and CoreGRID Network of Excellence and by Spanish CICYT under grant TIC2002-04498-C05-03 (Tracer).

ISSN 1433-8092

(see also [12]). Here, the game can be described by a Turing machine that can be used to compute the pay-off functions for the different games that arise when we consider different number of players as well as different number of resources. Thus, in contrast with our previous consideration here the game is described *uniformly* from a Turing machine.

This notion leads us to consider families of games that can be defined uniformly in the sense that there is a DTM working in polynomial time that gives the way of computing the utilities when the game is played with different number of players and/or different sets of actions. In this case we can again consider uniform families defined in *general* or *implicit* forms.

In this work we solve the fundamental question on the complexity of deciding the existence of a pure Nash equilibrium (the SPN problem) for strategic games. In the case of non-uniform game families we show that the SPN problem for games given in implicit and in general form is computationally hard in the first case Σ_2^p-complete and in the second NP-complete. When the game is given in standard normal form the SPN problem is tractable. The following table summarizes our results.

representation	Exist PNE?
implicit	Σ_2^p-complete
general	NP-complete
standard	AC^0
general with fixed number of players	P-complete

When we consider families of games defined uniformly and implicitly from a polynomial time deterministic Turing machine M, we show that the SPN problem is in Σ_2^p. Furthermore we show that there are Turing machines for which the problem is Σ_2^p-hard. Contrasting with this, when the representation of the games is in general form the positive and hardness results are for the NP class.

The paper is organized as follows: Section 2 contains basic definitions. In Section 3 we study the SPN problem for non-uniform families of games. Finally, Section 4 contains the results for uniform families of games.

2 Strategic games

The following is the mathematical definition of an strategic game borrowed form [14]

Definition 1. *An* strategic game Γ *is defined by the following components:*

- *A set of players denoted by* $N = \{1, \ldots, n\}$ *with* n *being the number of players.*
- *A finite set of actions* A_i *for each player* $i \in N$*. The elements of* $A_1 \times \ldots \times A_n$ *are called strategy profiles.*
- *An* utility *(or* payoff*) function* u_i *for each player* $i \in N$ *mapping* $A_1 \times \ldots \times A_n$ *to the integers.*

Given an strategy profile $a \in A_1 \times \ldots \times A_n$ and given any action $a_i \in A_i$, we denote by (a_{-i}, a_i) the strategy profile obtained by replacing the i-th component of a by a_i.

Definition 2. *An strategy profile* $a^* = (a_1^*, a_2^*, \ldots, a_n^*)$ *is an Strategic Pure Nash (*PNE*) equilibrium if, for any player* i *and any* $a_i \in A_i$ *we have* $u_i(a^*) \geq u_i(a_{-i}^*, a_i)$*.*

A Pure Nash equilibrium is an strategy profile in which no player can improve their utility by changing their action. Determining whether Pure Nash Equilibria exist is a problem that have attracted much research in computer science (see [16]). This problem can be formulated as follows:

Strategic Pure Nash (SPN)
Given an strategic game Γ, decide whether Γ has a Pure Nash equilibrium.

All through the paper we use standard notation for computational complexity classes. See for example [1, 2, 15].

3 Non-uniform families of games

In the context of computational complexity it is very important to define how an input game Γ is represented. In order to define an instance of the SPN problem we have to make clear how to describe the set of players, and for each player their set of actions and pay-off functions. Depending on the succinctness of the description of the action sets and depending on whether the pay-off functions are described by Turing Machines or tables we define three families of games that differ in their representation.

All the TMs appearing in the description of games are deterministic. We use the following convention: there is a pre-fixed interpretation of the contents of the output tape of a TM so that, both when the machine stops or when the machine is stopped, it always computes a value. Let also assume that Σ is a pre-fixed alphabet. Hence we can describe the pay-off functions of a game by giving a tuple $\langle M, 1^t \rangle$ where M is a deterministic TM (DTM) and t is a natural number bounding its computation time. The interpretation is that given an strategy profile a and a natural number i, the output of M on input $\langle a, i \rangle$ is the value of the pay-off function of the i-th player on input a.

First, we consider a way of describing the set of actions in which they are not given explicitly and directly, by listing all their actions, but succinctly and implicitly. We are interested in descriptions whose length does not depend dramatically on the number of the actions, but depends on the length of the actions. Such descriptions are exponentially more succinct than the sets they describe. The following definition captures this idea.

Strategic games in implicit form[1]. A game is a tuple $\Gamma = \langle 1^n, 1^m, M, 1^t \rangle$. This game has n players. For each player i, their set of actions is $A_i = \Sigma^m$ and $\langle M, 1^t \rangle$ is the description of the pay-off functions.

The second family of games is defined by considering that the set of actions of each player is given explicitly.

Strategic games in general form. A game is a tuple $\Gamma = \langle 1^n, A_1, \ldots, A_n, M, 1^t \rangle$. It has n players, for each player i, their set of actions A_i is given by listing all its elements. The description of their pay-off functions is given by $\langle M, 1^t \rangle$.

Finally we consider a less succinct way to describe games. This is the usual description adopted in basic books giving us a complete description in form of a bimatrix or trimatrix (set of bimatrices).

Strategic games in standard form. A game is a tuple $\Gamma = \langle 1^n, A_1, \ldots, A_m, T \rangle$. It has n players, and for each player i, their set of actions A_i is given explicitly. T is a table with an entry for each strategy profile a and a player i. In this case $u_i(a) = T(a, i)$.

[1] In the games in implicit form we assume $A_i = \Sigma^m$, this is not a major restriction because we can also consider $A_i \subseteq \Sigma^{\leq m}$ with just small modifications. In this case $\Gamma = \langle 1^n, 1^m, M_1, \ldots M_n, M, 1^t \rangle$ with $M_1, \ldots, M_n, M$ being DTM. The game is played by n players. For each player i, M_i is a succinct description of their set of actions $A_i \subseteq \Sigma^{\leq m}$. We say that $a_i \in A_i$ iff M_i accepts a_i in at most t steps. Given a and i, $u_i(a)$ is the output of $M(a, i)$ after at most t steps.

We analyze the complexity of the SPN problem in the different representations of games answering in this way the question posed by Papadimitriou in [16] for the case of strategic games. In the following we classify this problem and show that it is hard in all the representations for the matching complexity class. The exception is the case of the standard form and the general form when the number of players is constant when the problem can be solved in polynomial time. Before presenting our results we would like to introduce a gadget that we will use in all the constructions of games in the hardness proofs. The object is to associate an strategic game to a given property in such a way that the game has a PNE only in the case that the property is true.

Gadget game: Let P be a property. We associate to P the strategic game Γ_P defined as follows: It has players 1 and 2. Their action sets are the same, $A_1 = A_2 = \{0, 1\}$. And the pay-off functions are

$$u_1(a_1, a_2) = \begin{cases} 5 & \text{if } P \text{ is true,} \\ 4 & \text{if } P \text{ is false } \wedge a_1 = 0 \wedge a_2 = 1, \\ 3 & \text{if } P \text{ is false } \wedge a_1 = 1 \wedge a_2 = 1, \\ 2 & \text{if } P \text{ is false } \wedge a_1 = 1 \wedge a_2 = 0, \\ 1 & \text{if } P \text{ is false } \wedge a_1 = 0 \wedge a_2 = 0. \end{cases}$$

$$u_2(a_1, a_2) = \begin{cases} 5 & \text{if } P \text{ is true,} \\ 3 & \text{if } P \text{ is false } \wedge a_1 = 0 \wedge a_2 = 1, \\ 2 & \text{if } P \text{ is false } \wedge a_1 = 1 \wedge a_2 = 1, \\ 1 & \text{if } P \text{ is false } \wedge a_1 = 1 \wedge a_2 = 0, \\ 4 & \text{if } P \text{ is false } \wedge a_1 = 0 \wedge a_2 = 0. \end{cases}$$

Proposition 1. *Given a property P, the gadget game Γ_P has a* PNE *if and only if the property P is true.*

Proof. When P is true, since the utility of each player is equal to the maximum value 5 independently of their strategy, then no player has incentive to change their strategy. Hence, every strategy profile of Γ_P is a Nash equilibrium.

Now let us assume that P is false, whatever is the strategy profile (a_1, a_2), both players have an incentive to change their strategy:

$$\begin{array}{l} \text{if } a_1 = 0 \wedge a_2 = 0,\ u_1(0,0) < u_1(1,0) \\ \text{if } a_1 = 0 \wedge a_2 = 1,\ u_2(0,1) < u_2(0,0) \\ \text{if } a_1 = 1 \wedge a_2 = 0,\ u_2(1,0) < u_2(1,1) \\ \text{if } a_1 = 1 \wedge a_2 = 1,\ u_1(1,1) < u_1(0,1). \end{array}$$

Hence, no strategy profile of Γ_P is a Nash equilibrium. □

First we study the complexity of deciding whether a game in implicit form has a PNE. We show that this problem is really hard since it is complete for the second level of the Polynomial Hierarchy. Observe that the proof of Theorem 3.4 of [10] can be rewritten to show that the problem of deciding whether a given strategy (of a game given in implicit form) is a Nash equilibrium is coNP-complete. At first glance this fact seems to imply that the hardness of the SPN problem follows trivially from the coNP-completeness and the additional existential quantification. It is worth noticing that this approach is false in general as it is known that the equivalence problem for circuits is coNP-complete while the isomorphisms for circuits is not Σ_2^p-hard unless the polynomial hierarchy collapses to the third level [17].

Theorem 1. *The* SPN *problem for strategic games in implicit form is* Σ_2^p*-complete.*

Proof. Let $\Gamma = \langle 1^n, 1^m, M, 1^t \rangle$ be an strategic game in implicit form, the problem of deciding whether Γ has a PNE can be formalized as follows:

$$\Gamma \in \text{SPN} \Leftrightarrow \exists a_1^* \in A_1 \ \dots \exists a_n^* \in A_n \ \forall a_1 \in A_1 \ \dots \forall a_n \in A_n$$
$$u_1(a_{-1}^*, a_1) \leq u_1(a_{-1}^*, a_1^*) \wedge \dots \wedge u_n(a_{-n}^*, a_n) \leq u_n(a_{-n}^*, a_n^*).$$

Hence we can define an Alternating Turing machine that guesses the strategy profile $(a_1^*, \dots, a_n^*)$ and then using a universal state it can verify that this strategy profile is a Nash equilibrium. Since the length of any action is bounded by m, and for each player i, u_i can be computed in time t, then the computation time of this Alternating Turing machine is bounded by a polynomial with respect to $\max\{n, m, t\}$. Then SPN $\in \Sigma_2^p$.

In order to prove the hardness of the SPN problem let us consider a restricted version of the Quantified Boolean Formula, the Q2SAT problem, which is Σ_2^p-complete. Recall that Q2SAT is defined as follows:

Given $\Phi = \exists \alpha_1, \dots, \alpha_{n_1} \forall \beta_1, \dots \beta_{n_2} F$ where F is a Boolean formula over the boolean variables $\alpha_1, \dots, \alpha_{n_1}, \beta_1, \dots, \beta_{n_2}$, decide whether Φ is valid.

For each Φ we define a game $\Gamma(\Phi)$ as follows. There are four players:

- Player 1, the *existential player*, assigns truth values to the boolean variables $\alpha_1, \dots, \alpha_{n_1}$. Their set of actions is $A_1 = \{0,1\}^{n_1}$ and $a_1 = (\alpha_1, \dots \alpha_{n_1}) \in A_1$.
- Player 2, the *universal player*, assigns truth values to the boolean variables $\beta_1, \dots, \beta_{n_2}$ and then their set of actions is $A_2 = \{0,1\}^{n_2}$ and $a_2 = (\beta_1, \dots, \beta_{n_2}) \in A_2$.
- Players 3 and 4 avoid entering into a Nash equilibrium when the actions played by players 1 and 2 do not satisfy F. Their set of actions are $A_3 = A_4 = \{0,1\}$.

Let us denote by $F(a_1, a_2)$ the truth value of F under the assignment given by a_1 and a_2. Now it only remains to define the utility functions in such a way that they guarantee that Φ is valid if and only if $\Gamma(\Phi)$ has a Nash equilibrium.

$$u_1(a_1, a_2, a_3, a_4) = \begin{cases} 1 & \text{if } F(a_1, a_2) = 1, \\ 0 & \text{otherwise.} \end{cases}$$

$$u_2(a_1, a_2, a_3, a_4) = \begin{cases} 1 & \text{if } F(a_1, a_2) = 0, \\ 0 & \text{otherwise.} \end{cases}$$

$$u_3(a_1, a_2, a_3, a_4) = \begin{cases} 5 & \text{if } F(a_1, a_2) = 1, \\ 4 & \text{if } F(a_1, a_2) = 0 \wedge a_3 = 0 \wedge a_4 = 1, \\ 3 & \text{if } F(a_1, a_2) = 0 \wedge a_3 = 1 \wedge a_4 = 1, \\ 2 & \text{if } F(a_1, a_2) = 0 \wedge a_3 = 1 \wedge a_4 = 0, \\ 1 & \text{if } F(a_1, a_2) = 0 \wedge a_3 = 0 \wedge a_4 = 0. \end{cases}$$

$$u_4(a_1,a_2,a_3,a_4) = \begin{cases} 5 & \text{if } F(a_1,a_2)=1, \\ 3 & \text{if } F(a_1,a_2)=0 \wedge a_3=0 \wedge a_4=1, \\ 2 & \text{if } F(a_1,a_2)=0 \wedge a_3=1 \wedge a_4=1, \\ 1 & \text{if } F(a_1,a_2)=0 \wedge a_3=1 \wedge a_4=0, \\ 4 & \text{if } F(a_1,a_2)=0 \wedge a_3=0 \wedge a_4=0. \end{cases}$$

Finally, we claim: Φ is valid $\Leftrightarrow$ $\Gamma(\Phi)$ has a PNE.

Let us assume that Φ is valid. Then there exists $a_1 \in \{0,1\}^{n_1}$ such that for all $a_2 \in \{0,1\}^{n_2}$, $F(a_1,a_2) = 1$. In terms of the game $\Gamma(\Phi)$ this means that if player 1 plays action a_1, for all $a_2 \in A_2$, $a_3 \in A_3$, and $a_4 \in A_4$, player 1 has no incentive to change their action: $u_1(a_1,a_2,a_3,a_4) \geq u_1(a_1',a_2,a_3,a_4)$ for any $a_1' \in A_1$. Moreover, for any $a_2 \in A_2$ we have that $u_2(a_1,a_2,a_3,a_4) = 0$. Then player 2 neither has incentive to change her strategy when player 1 selects a_1. Finally, since $F(a_1,a_2) = 1$, then $u_3(a_1,a_2,a_3,a_4) = u_4(a_1,a_2,a_3,a_4) = 5$. for any $a_3 \in A_3$ and for any $a_4 \in A_4$. Hence, given a_1, if we fix the values of a_2, a_3 and a_4 arbitrarily, no player has incentive to change her strategy. There will be at least as many pure Nash equilibria as the number of elements of $A_2 \times A_3 \times A_4$.

Now let us assume that Φ is not valid. It means that for any $a_1 \in A_1$ there exists $a_2 \in A_2$ such that $F(a_1,a_2) = 0$. In the case that $F(a_1,a_2) = 0$, we have that players 3 and 4 play the same role than players 1 and 2 of the gadget game Γ_P being P equal $F(a_1,a_2) = 1$. Since P is false players 3 and 4 always have incentive to change their strategies as we have shown in the proof of proposition 1. Hence, if $F(a_1,a_2) = 0$ then for any $a_3 \in A_3$ and any $a_4 \in A_4$, (a_1,a_2,a_3,a_4) is not a Nash equilibrium in this case.

If $F(a_1,a_2) = 1$, since Φ is not valid, there exists $a_2' \in A_2$ such that $F(a_1,a_2') = 0$. Therefore player 2 has incentive to change her strategy a_2 to a_2' because $u_2(a_1,a_2,a_3,a_4) = 0$ and $u_2(a_1,a_2',a_3,a_4) = 1$. But the strategy profile (a_1,a_2',a_3,a_4) is not a PNE either, as we have seen before.

It remains to show that a description of the above game in implicit form can be obtained in polynomial time. Note that the number of players is 4 and their set of actions can be described succinctly with respect their length. It only remains to show that we can construct a DTM that computes their pay-off functions in polynomial time with respect to number of variables of F.

Consider the following TM (which depends on F):

```
input a1, a2, a3, a4, i
if Eval(F, a1, a2)
   then
      if i < 3 then u = 1 else u = 5 end if
   else
      if (i = 3 ∧ a3 = 0 ∧ a4 = 1) ∨ (i = 4 ∧ a3 = 0 ∧ a4 = 0) then u = 4 end if
      if (i = 3 ∧ a3 = 1 ∧ a4 = 1) ∨ (i = 4 ∧ a3 = 0 ∧ a4 = 1) then u = 3 end if
      if (i = 3 ∧ a3 = 1 ∧ a4 = 0) ∨ (i = 4 ∧ a3 = 1 ∧ a4 = 1) then u = 2 end if
      if (i = 3 ∧ a3 = 0 ∧ a4 = 0) ∨ (i = 4 ∧ a3 = 1 ∧ a4 = 0) then u = 1 end if
      if i < 3 then u = 0 end if
end if
output u
```

where Eval is a TM that evaluates a CNF formula in time $O((n + |F|)^2$. The overall running time of the machine can be bounded by $(n + |F|)^3$. Furthermore its representation can be computed in polynomial time from F. □

In the previous proof, for the sake of clarity, the number of players has been selected to be four. A similar game with only two players can be easily defined.

Given a formula $\Phi = \exists\alpha_1, \dots, \alpha_{n_1} \forall\beta_1, \dots \beta_{n_2} F$ where F is a Boolean formula over the boolean variables $\alpha_1, \dots, \alpha_{n_1}, \beta_1, \dots, \beta_{n_2}$, we define a game $\Gamma(\Phi)$ as follows. There are two players:

- Player 1, the *existential player*, assigns truth values to the boolean variables $\alpha_1, \dots, \alpha_{n_1}$ and sets the value of the additional bit a_3. Their set of actions is $A_1 = \{0, 1\}^{n_1+1}$.
- Player 2, the *universal player*, assigns truth values to the boolean variables $\beta_1, \dots, \beta_{n_2}$ and sets the value of the additional bit a_4. Then their set of actions is $A_2 = \{0, 1\}^{n_2+1}$.

Let us denote by $F(a_1, a_2)$ the truth value of F under the assignment given by $a_1 = (\alpha_1, \dots, \alpha_{n_1})$ and $a_2 = (\beta_1, \dots, \beta_{n_2})$. Now it only remains to define the utility functions in such a way that they guarantee that Φ is valid if and only if $\Gamma(\Phi)$ has a Nash equilibrium.

$$u_1(a_1, a_2, a_3, a_4) = \begin{cases} 5 & \text{if } F(a_1, a_2) = 1, \\ 4 & \text{if } F(a_1, a_2) = 0 \wedge a_3 = 0 \wedge a_4 = 1, \\ 3 & \text{if } F(a_1, a_2) = 0 \wedge a_3 = 1 \wedge a_4 = 1, \\ 2 & \text{if } F(a_1, a_2) = 0 \wedge a_3 = 1 \wedge a_4 = 0, \\ 1 & \text{if } F(a_1, a_2) = 0 \wedge a_3 = 0 \wedge a_4 = 0. \end{cases}$$

$$u_2(a_1, a_2, a_3, a_4) = \begin{cases} 5 & \text{if } F(a_1, a_2) = 1, \\ 3 & \text{if } F(a_1, a_2) = 0 \wedge a_3 = 0 \wedge a_4 = 1, \\ 2 & \text{if } F(a_1, a_2) = 0 \wedge a_3 = 1 \wedge a_4 = 1, \\ 1 & \text{if } F(a_1, a_2) = 0 \wedge a_3 = 1 \wedge a_4 = 0, \\ 4 & \text{if } F(a_1, a_2) = 0 \wedge a_3 = 0 \wedge a_4 = 0. \end{cases}$$

Each player controls with the additional bit the corresponding gadget game. Therefore we obtain the same hardness result for the case of two players.

Corollary 1. *The* SPN *problem for games in implicit form with k players is Σ_2^p-complete, for any $k \geq 2$.*

Contrasting with the previous results, when we allow to describe the set of actions explicitly, although the SPN problem remains hard, it is not as hard as the SPN problem for games where the set of actions are described implicitly.

Theorem 2. *The* SPN *problem for strategic games in general form is NP-complete.*

Proof. Consider $\Gamma = \langle 1^n, A_1, \dots, A_n, M, 1^t \rangle$, We can conjecture a strategic profile $(a_1^*, \dots, a_n^*)$ and then check that for any i and any $a_i \in A_i$ $u_i(a^*) \geq u_i(a_{-i}^*, a_i)$. Each computation of M takes time at most t and the overall number of tests to be performed is at most $\sum_{i=1}^n |A_i|$. As the sets of actions are given explicitly the Nash equilibrium property can be checked in time polynomial in the input size.

In order to prove the hardness let us reduce the Satisfiability of boolean formulae problem to the SPN problem in general form. Given a formula F in conjunctive normal form on n variables with $n \geq 2$, we consider the game $\Gamma(F)$ defined as follows: We have n players, for each $1 \leq i \leq n$, $A_i = \{0,1\}$. Therefore the set of strategic profiles coincides with the set of truth assignments. The utilities are defined as follows:

$$u_1(a) = \begin{cases} 5 & \text{if } F(a) = 1, \\ 4 & \text{if } F(a) = 0 \wedge a_1 = 0 \wedge a_2 = 1, \\ 3 & \text{if } F(a) = 0 \wedge a_1 = 1 \wedge a_2 = 1, \\ 2 & \text{if } F(a) = 0 \wedge a_1 = 1 \wedge a_2 = 0, \\ 1 & \text{if } F(a) = 0 \wedge a_1 = 0 \wedge a_2 = 0, \end{cases} \quad u_2(a) = \begin{cases} 5 & \text{if } F(a) = 1, \\ 4 & \text{if } F(a) = 0 \wedge a_1 = 0 \wedge a_2 = 0, \\ 3 & \text{if } F(a) = 0 \wedge a_1 = 0 \wedge a_2 = 1, \\ 2 & \text{if } F(a) = 0 \wedge a_1 = 1 \wedge a_2 = 1, \\ 1 & \text{if } F(a) = 0 \wedge a_1 = 1 \wedge a_2 = 0. \end{cases}$$

And, for any $j > 2$

$$u_j(a) = \begin{cases} 5 & \text{if } F(a) = 1, \\ 1 & \text{otherwise.} \end{cases}$$

Let us show that F is satisfiable iff $\Gamma(F)$ has a PNE. If F is satisfiable, any strategy profile that satisfies F is a PNE since all the players have the maximum utility value when they play this strategy profile. If F is not satisfiable, then no player i, for $3 \leq i \leq n$ has any incentive to change their strategy, the utility value is always equal to 0. But now players 1 and 2 play the same role of the players of the gadget game Γ_P being P equal to $(\exists a\ F(a) = 1)$. Since such a does not exist, then P is false and then players 1 and 2 always have incentive to change their strategy. Hence, in this case $\Gamma(F)$ does not have a PNE.

Notice that $\Gamma(F)$ can be represented in general form by $\langle 1^n, \{0,1\} \ldots \{0,1\}, M^F, 1^{(n+|F|)^3} \rangle$ where M^F is a TM that on input (a, i), evaluates the formula F on input a. Afterwards it implements the utility function of the i-th player. Since we can construct M^F in polynomial time and its computation time is also polynomial, always respect to $|F|$, we have that the representation of $\Gamma(F)$ in general form can be constructed in polynomial time with respect to $|F|$. □

The hardness in the above proof is obtained by constructing a game in which the number of allowed actions for each player is two. Hence we obtain the same hardness result even in the restricted case of considering a two actions for each player.

Corollary 2. *The* SPN *problem for strategic games in general form is NP-complete, even in the case that the number of actions of each a player is some constant k, for any $k \geq 2$.*

Contrasting with the previous hardness results, by analyzing the case of strategic games in general form when the number of players is fixed, we can see that the SPN problem becomes tractable.

Theorem 3. *For any $k \geq 2$, the* SPN *problem for strategic games in general form with k-players is P-complete.*

Proof. For the membership part let us consider a game $\Gamma_k = \langle A_1, \ldots A_k, M, 1^t \rangle$ with k players, being k a constant. We assume for simplicity that each A_i has m actions each one coded in $O(\log m)$ bits, therefore Γ_k has size $O(km \log m + t + |M|)$. Since the number of strategies for each player is at most m and the number of players is a constant k, verify that an strategy profile of Γ_k is a PNE can be done in time $O(tmk)$. Hence we can construct a DTM such that on input $\langle A_1, \ldots A_k, M, 1^t \rangle$ verifies whether at least one of their strategy profiles is a PNE. Since there are at most m^k strategy profiles the total computation time of this machine is $O(m^k tmk)$.

For the hardness we consider the Circuit Value Problem (CVP). Recall that its instances are pairs $\langle C, x\rangle$ being C a description of a boolean circuit with n input gates and one output gate, and x an assignment of $0,1$ values to C's input gates. Let us consider the gadget game Γ_P considering that P is equal to $(C(x) = 1)$, i.e. the circuit C on input x evaluates to 1. Hence by the proposition 1 we have that $\langle C, x\rangle \in$ CVP iff Γ_P has a PNE.

It remains to show that the general representation of the game Γ_P can be computed in logarithmic space with respect to the length of $\langle C, x\rangle$. Note that we can represent Γ_P in general form as $\langle\{0,1\},\{0,1\}, M, 1^t\rangle$, where M is the following TM:

```
input a1, a2, i
if EvalC(C, x)
   then output 5
   else
      if (i = 1 ∧ a1 = 0 ∧ a2 = 1) ∨ (i = 2 ∧ a1 = 0 ∧ a2 = 0) then output 4 end if
      if (i = 1 ∧ a1 = 1 ∧ a2 = 1) ∨ (i = 2 ∧ a1 = 0 ∧ a2 = 1) then output 3 end if
      if (i = 1 ∧ a1 = 1 ∧ a2 = 0) ∨ (i = 2 ∧ a1 = 1 ∧ a2 = 1) then output 2 end if
      if (i = 1 ∧ a1 = 0 ∧ a2 = 0) ∨ (i = 2 ∧ a1 = 1 ∧ a2 = 0) then output 1 end if
end if.
```

where EvalC is a polynomial time TM that on input $\langle C, x\rangle$ computes $C(x)$ in $O(n^2)$ time and the time allowed is $t = |\langle C, x\rangle|^3$.

Hence, it is easy to see that the representation of Γ_P in general form can be computed in logarithmic space with respect to $|\langle C, x\rangle|$. □

Finally, we provide lower complexity bounds for the SPN problem when games are given in standard form.

Lemma 1. *The SPN problem for strategic games in standard form is in AC^0.*

Proof. In order to prove that the SPN problem is in AC^0 we show that the property that defines the problem can be expressed as a formula in First Order Logic in the following way:

$$\text{SPN} \equiv \bigvee_{a^* \in A_1 \times \ldots \times A_n} \text{IS-SPN}(a^*),$$

where IS-SPN$(a^*) \equiv$ (a^* is a PNE). This last predicate can be expressed as

$$\text{IS-SPN}(a^*) = \bigwedge_{i=1}^{n} \text{BR}(a^*, i),$$

where $\text{BR}(a^*, i) \equiv (\forall a_i \in A_i\ u_i(a^*_{-i}, a^*_i) \geq u_i(a^*_{-i}, a_i))$. Hence, we can express BR as follows:

$$\text{BR}(a^*, i) = \bigwedge_{a_i \in A_i} \text{GEQ}(a^*, i, a_i) \quad \text{and}$$

$$\text{GEQ}(a^*, i, a_i) = \bigvee_{1 \leq k \leq s} \Big(\bigwedge_{k < l \leq s} (u_i(a^*_{-i}, a^*_i)[l] = u_i(a^*_{-i}, a_i)[l]) \wedge u_i(a^*_{-i}, a^*_i)[k] = 1 \wedge u_i(a^*_{-i}, a_i)[k] = 0 \Big),$$

where s is the maximum length of the utility values and $u_i(a)[k]$ denotes the k-th bit of $u_i(a)$. □

4 Uniform families of strategic games with polynomial time computable utilities

In the previous section we have analyzed the representations of the strategic games as potential inputs of the SPN problem. Here we are interested in families of strategic games that arise when the utility functions are computable in polynomial time. Thus we are interested in families of games defined uniformly by Turing machines. But, what does exactly mean that a game has polynomial time computable pay-off functions? Even though in many papers studying the computational complexity of some specific games, it is assumed that the utilities are computable in polynomial time (see for example [6, 10, 7–9]) and this assumption has had different interpretations.

For instance, Gottlob et al. consider that "each player has a a polynomial time computable real valued utility function" however a machine computing such function is not given as part of the description of a game [10]. Fotakis et al. [6] consider congestion games with a different representation. A congestion game is defined by n players, a set E of resources, and a delay function d mapping $E \times \{1, \ldots, n\}$ to the integers. The action for each player are subsets of E. The pay-off functions can be computed as follows:

$$u_i(a_1, \ldots, a_n) = -(\sum_{e \in a_i} d(e, f(a_1, \ldots, a_n, e)))$$

being $f(a_1, \ldots, a_n, e) = |\{i \mid e \in a_i\}|$.

In the second case, they consider a uniform family of games in the sense that the different instances are given by considering different number of players, action sets and delay functions, but in each of them the pay-off functions can be computed by a DTM which works in polynomial time with respect to n and m, being m the maximum length of the actions a_i.

Following the same spirit of Fotakis et al., for each DTM M we define uniform families of strategic games in such a way that the pay-off functions of each game in the family are computed by M. Moreover, as in the previous section, we consider further refinements according to the input representation.

Let M be a DTM and let us assume that an alphabet Σ is fixed. We define the following uniform families of games associated to M:

M-implicit form family[2]. It is an implicit description of the family of games in which the pay-off functions are computed by the DTM M. Each instance of the family specifies the number of players n and their set of actions in an succinct way. We consider that a description of a set is succinct when the length of the description is at most polynomial with respect to its length. Formally, the M-implicit form family is defined as follows:

$$\{\langle 1^n, 1^{m_1}, \ldots, 1^{m_n} \rangle \mid n, m_1, \ldots, m_n \in \mathbb{N}\}.$$

In the game described by $\langle 1^n, 1^m, M_1, \ldots, M_n \rangle$, if a is an strategy profile of such game, and $1 \leq i \leq n$, then the utility of the i-th player on a is defined as $u_i(a) = M(a, i)$.

M-general form family. It is a general form description of the family of games in which the pay-off functions are computed by M. Each instance of the family describes a game by giving the number of players n and the set of actions of each player. Here, every set of

[2] In the games in implicit form we assume $A_i = \Sigma^{\leq m_i}$. We can also consider $A_i \subseteq \Sigma^{\leq m_i}$. In this case the machine M has to be able to recognize whether a given action a_i belongs to A_i.

actions is given by listing all its elements. Formally, the M-general form family is defined as follows:

$$\{\langle 1^n, A_1, \ldots, A_n\rangle \mid n, m \in \mathbb{N} \wedge \forall i\ A_i \text{ is given by listing all its elements in } \Sigma^*\}$$

As in the M-implicit form, in the game described by $\langle 1^n, A_1, \ldots, A_n\rangle$, if a is an strategy profile of such game, and $1 \le i \le n$, then the utility of the i-th player on a is defined as $u_i(a) = M(a, i)$.

Hence, given a family of games defined from a polynomial time DTM M, we can also pose the question of determining whether a game of this family has a Nash equilibrium.

M-Strategic Pure Nash (M-SPN)
Given an strategic game Γ, whose pay-off functions are defined by M, decide whether Γ has a Pure Nash equilibrium.

As we have seen in the previous section, depending on whether the games are described in implicit or general form we obtain different hardness results. In the following we show that the M-SPN problem for the implicit form games is Σ_2^p-complete for a particular polynomial time DTM M, while the M'-SPN problem for the general form games is NP-complete for another polynomial time DTM M'. Since we are considering uniform families of games, the main difference with respect to the proofs of the analogous results in the previous section is that the DTM computing the utilities of the game defined in each one of the reductions is not parameterized by the quantified boolean formula Φ in the case of the Σ_2^p-hardness results, or by the boolean formula F in the case of the NP-hardness results. Now these formulae will be part of the input of the machines as a strategy for some player.

Theorem 4. *There exists a polynomial time* DTM M *for which the* M-SPN *problem for games in the* M*-implicit form family is* Σ_2^p*-complete.*

Proof. Following the same arguments of the membership proof of theorem 1, for any fixed polynomial time DTM M, the problem of deciding whether a game Γ in M-implicit form has a PNE can be solved by an Alternating TM, with 2 alternations, existential and universal, in polynomial time. Hence M-SPN $\in \Sigma_2^p$.

In order to prove the hardness, we have to define first the polynomial time DTM M. Let M be the TM such that on input $(\Phi, a_1, a_2, a_3, a_4, i)$ being $\Phi = \exists \alpha_1, \ldots, \alpha_{n_1} \forall \beta_1, \ldots \beta_{n_2} F$ an instance of the Q2SAT problem, $a_1 \in A_1 = \{0,1\}^{n_1}$, $a_2 \in A_2 = \{0,1\}^{n_2}$ and $a_3, a_4 \in \{0,1\}$, computes the utilities defined in the proof of theorem 1, but now we consider that the quantified boolean formula Φ is an element of the input. It is easy to see that M works in polynomial time with respect to the input length.

Once we have defined M, we can show that Q2SAT can be reduced to M-SPN in implicit form. For each Φ we define a game $\Gamma(\Phi)$ with five players. Players 1, 2, 3 and 4 are defined exactly equal to the four players of the game defined in the theorem 1. The difference is that now we have an additional player, player 0 who has a unique action that defines the rules of the game, i.e. $A_0 = \{\Phi\}$. As we have shown in theorem 1, Φ is valid if and only if $\Gamma(\Phi)$ has a PNE, and the description of $\Gamma(\Phi)$ in implicit form can be obtained in polynomial time. □

From the previous proof it is easy to show that for any polynomial time DTM M the M-SPN problem for games in M-implicit form is in Σ_2^p.

Corollary 3. *For any polynomial time* DTM M*, the* M-SPN *problem for games in the* M*-implicit form family is in* Σ_2^p.

Using similar arguments we can modify the proof of theorem 2 in order to obtain the following result. We only have to consider an additional player whose set of actions contains only the input formula for the reduction.

Theorem 5. *There exists a polynomial time* DTM M *for which the* M-SPN *problem for games in the* M*-general form family is* NP*-complete.*

Corollary 4. *For any polynomial time* DTM M*, the* M-SPN *problem for games in the* M*-general form family is in* NP.

If we consider the results presented in [10], they propose to study, among many other problems, the complexity of the SPN problem for games in

$$\bigcup_{M \in \mathsf{polyTM}} M\textbf{-general form family},$$

where polyTM is the class of TM working in polynomial time. They assume that the utility functions of their games are polynomially computable functions and they show that deciding whether a game in general form has a PNE is NP-complete. To prove the membership in NP, they strongly need to make use of the assumption that the utilities are polynomial time computable. However, in their hardness result, they construct polynomial time computable utilities, but the utilities are non-uniform in the sense that for each instance they get a different utility function.

Our contribution is different, for the uniform families our reduction produce a Turing machine for all the game instances. Furthermore, in the previous section, for non-uniform families of games, we give a general way of describing all the games with "computable utilities". In order to prove our complexity results, we do not have to assume that the description of the pay-off functions can be given as polynomial time DTM, we represent any 'computable' pay-off function by giving a DTM and a natural number t (in unary) bounding its computation time.

References

1. J. L. Balcazar, J. Díaz, and J. Gabarró. *Structural Complexity I.* Springer-Verlag, Heidelberg, 1988.
2. J. L. Balcazar, J. Díaz, and J. Gabarró. *Structural Complexity II.* Springer-Verlag, Heidelberg, 1990.
3. E. Ben-Porath. The complexity of computing a best response automaton in repeated games with mixed strategies. *Games and Economic Behavior*, pages 1–12, 1990.
4. F. Chu and J. Halpern. On the NP-completeness of finding an optimal strategy in games with commons pay-offs. *International Journal of Game Theory*, 2001.
5. V. Conitzer and T. Sandholm. Complexity results about nash equilibra. In *Proceedings IJCAI 2003*, pages 765–771, 2003.
6. A. Fabrikant, C. Papadimitriou, and K. Talwar. The complexity of pure nash equilibria. In *STOC 2004*, pages 604–612, 2004.
7. D. Fotakis, S. Kontogiannis, E. Koutsoupias, M. Mavronicolas, and P. Spirakis. The structure and complexity of nash equilibria for a selfish routing game. In *ICALP 2002*, pages 123–134, 2002.
8. D. Fotakis, S. Kontogiannis, and P. Spirakis. Selfish unsplittable flows. In *ICALP 2004*, pages 593–605, 2004.
9. M. Gairing, T. Lücking, M. Mavronicolas, B. Monien, and M. Rode. Nash equilibria in discrete routing games with convex latency functions. In *ICALP 2004*, pages 645–657, 2004.
10. G. Gottlob, G. Greco, and F. Scarcello. Pure nash equilibria: Hard and easy games. In *Theoretical Aspects of Rationality and Knowledge, Proceedings.*, pages 215–230, 2003.
11. D. Koller and M-Megiddo. The complexity of two-person zero sum games in extensive form. *Games and Economic Behavior*, pages 528–552, 1992.

12. S. Kontogiannis and P. Spirakis. Atomic selfish routing in networks: A survey. Technical report, FLAGS-TR-04-79, 2004.
13. M. Littman and P. Stone. A polynomial time nash equilibrium for repeated games. *ACM-EC*, pages 48–54, 2003.
14. M. Osborne and A. Rubinstein. *A Course in Game Theory*. MIT Press, 1994.
15. C. Papadimitriou. *Computational Complexity*. Addison-Wesley, Reading, Mass., 1994.
16. C. Papadimitriou. Algorithms, games and the internet. In *STOC 2001*, pages 4–8, 2001.
17. T. Thierauf. *The computational complexity of equivalence and isomorphisms problems*. Springer, 2000.

ECCC ISSN 1433-8092
http://www.eccc.uni-trier.de/eccc
ftp://ftp.eccc.uni-trier.de/pub/eccc
ftpmail@ftp.eccc.uni-trier.de, subject 'help eccc'