Pure Nash Equilibria: Hard and Easy Games

Georg Gottlob
Inst. für Informationssysteme
Technische Universität Wien
A-1040 Vienna, Austria
gottlob@dbai.tuwien.ac.at

Gianluigi Greco
DEIS
Università della Calabria
I-87030 Rende, Italy
ggreco@si.deis.unical.it

Francesco Scarcello
DEIS
Università della Calabria
I-87030 Rende, Italy
scarcello@deis.unical.it

Abstract

In this paper we investigate complexity issues related to pure Nash equilibria of strategic games. We show that, even in very restrictive settings, determining whether a game has a pure Nash Equilibrium is NP-hard, while deciding whether a game has a strong Nash equilibrium is Σ_2^P-complete. We then study practically relevant restrictions that lower the complexity. In particular, we are interested in quantitative and qualitative restrictions of the way each player's move depends on moves of other players. We say that a game has small neighborhood if the utility function for each player depends only on (the actions of) a logarithmically small number of other players. The dependency structure of a game $\mathcal{G}$ can be expressed by a graph $G(\mathcal{G})$ or by a hypergraph $H(\mathcal{G})$. Among other results, we show that if $\mathcal{G}$ has small neighborhood and if $H(\mathcal{G})$ has bounded hypertree width (or if $G(\mathcal{G})$ has bounded treewidth), then finding pure Nash and Pareto equilibria is feasible in polynomial time. If the game is graphical, then these problems are LOGCFL-complete and thus in the class $\mathcal{NC}_2$ of highly parallelizable problems.

1 Introduction and Overview of Results

The theory of strategic games and Nash equilibria has important applications in economics and decision making [31, 2]. Determining whether Nash equilibria exist, and effectively computing them, are relevant problems that have attracted much research in computer science (see, e.g., [7, 28, 25]). Most work has been dedicated to complexity issues related to *mixed equilibria* of games with *mixed strategies*, where the player's choices are not deterministic and are regulated by probability distributions. In that context, the existence of a Nash equilibrium is guaranteed by Nash's famous theorem [31], but it is currently open whether such an equilibrium can be computed in polynomial time, cf. [37]. First results on the computational complexity for a two-person game have been presented by Gilboa and Zemel [12], while extensions to more general types of games have been provided by Megiddo and Papadimitriou [29], and by Papadimitriou [35]. A recent paper of Conitzer and Sandholm [6] also proved the NP-hardness of determining whether Nash equilibria with certain natural properties exist. In the present paper, we are not dealing with mixed strategies, but rather investigate the complexity of deciding whether there exists a Nash equilibrium in the case of *pure strategies*, where each player chooses to play an action in a deterministic, non-aleatory manner. Nash equilibria for pure strategies are briefly referred to as *pure Nash equilibria*. Note that in the setting of pure strategies, a pure Nash equilibrium is not guaranteed to exist (see, for instance, Osborne and Rubinstein [32]). Particular classes of games having pure Nash equilibria have been studied by Rosenthal [39], Monderer and Shapley [30], and, more recently, by Fotakis et al. [10].

Our goal is to study fundamental questions such as the *existence* of pure Nash, Pareto, and strong Nash equilibria, the *computation* of such equilibria, and to find arguably realistic restrictions under which these problems become tractable. Throughout the paper, Pareto and strong Nash equilibria are considered only in the setting of pure strategies.

While pure strategies are conceptually simpler than mixed strategies, the associated computational problems appear to be harder. In fact, we show that even under very limitative restrictions on the set of allowed strategies, determining whether a game has a pure Nash or Pareto Equilibrium is NP-complete, while deciding whether a game has a strong Nash equilibrium is even Σ_2^P-complete. However, by jointly applying suitable pairs of more realistic restrictions, we obtain settings of practical interest in which the complexity of the above problems is drastically reduced. In particular, determining the existence of a pure Nash equilibrium and computing such an equilibrium will be feasible in polynomial time and we will show that, in certain cases, these problems are even complete for the very low complexity class LOGCFL, which means that these problems are essentially as easy as the membership problem for context-free languages, and are thus highly parallelizable (in NC_2).

In the setting of pure strategies, to which we will restrict our attention in the sequel of this paper, a finite strategic game is one in which each player has a finite set of possible actions, from which she chooses an action once and for all, independently of the actual choices of the other players. The choices of all players can thus be thought to be made simultaneously. The choice of an action by a player is referred to as the player's *strategy*. It is assumed that each player has perfect knowledge over all possible actions and over the possible strategies of all players. A *global strategy* consists of a tuple containing a strategy for each player. Each player has a polynomial-time computable real valued *utility function*, which allows her to assess her subjective utility of each possible global strategy (global strategies with higher utility are better). A pure *Nash equilibrium* [31] is a global strategy in which no player can improve her utility by changing her action (while the actions of all other players remain unchanged). A *strong Nash equilibrium* [1] is a pure Nash equilibrium where no change of strategies of whatever coalition (i.e., group of players) can simultaneously increase the utility for all players in the coalition. A pure Nash equilibrium is *Pareto optimal* (see, e.g., [27]) if the game admits no other pure Nash equilibrium for which each player has a strictly higher utility. A Pareto-optimal Nash equilibrium is also called a Pareto Nash Equilibrium.

Before describing our complexity results, let us discuss various parameters and features that will lead to restricted versions of strategic games. We consider restrictions of strategic games which impose quantitative and/or qualitative limitations on how the decisions of an agent may be influenced by the other agents. This is one way of capturing the idea of bounded rationality in strategic games. For a different view of bounded rationality, see also Papadimitriou and Yannakakis [34].

The *set of neighbors* $Neigh(p)$ of a player is the set of other players who potentially matter w.r.t. p's utility function. Thus a player $q \neq p$ is not in $Neigh(p)$ iff p's utility function does not directly depend on the actions of q. We assume that each game is equipped with a polynomial-time computable function $Neigh$ with the above property.[1]

A first idea towards the identification of tractable classes of games is to restrict the cardinality of $Neigh(p)$ for all players p. For instance, consider a set of companies in a market. Each company has usually a limited number of other market players on which it bases its strategic decisions. These relevant players are usually known and constitute the neighbors of the company in our setting. However, the game outcome still depends on the interaction of all players, though possibly in an indirect way. Indeed, the choice of a company influences the choice of its competitors, and hence, in turn, the choice of competitors of its competitors, and so on. In this more general setting, a number of real-world cases can be modelled in a very natural way.

[1]Note that each game can be trivially extended to this setting by letting $Neigh(p)$ be the set of all players for each player p. In most cases, however, one will be able to provide a much better neighborhood function.

We can thus define the concept of *bounded neighborhood* as follows:

Bounded Neighborhood: Let $k > 0$ be a fixed constant. A strategic game with associated neighborhood function $Neigh$ has *k-bounded neighborhood* if, for each player p, $|Neigh(p)| \leq k$. While in some setting the bounded neighborhood assumption is realistic, in other settings the constant bound appears to be too harsh an imposition. Let $Act(p)$ be the set of possible actions of a player p, and let us denote by $||\mathcal{G}||$ the total size of the description of a game $\mathcal{G}$ (i.e., the input size n). We define the notion of *small neighborhood* as follows:

Small Neighborhood: A class of strategic games has *small neighborhood* if, for the games $\mathcal{G}$ in this class and for the players p of $\mathcal{G}$,

$$|Neigh(p)| = O(\frac{\log ||\mathcal{G}||}{\log |Act(p)|}).$$

In other words, the class has small neighborhood if there is a constant c such that for all but finitely many pairs (G, p) of games and players, $|Neigh(p)| < c \times (\frac{\log |\mathcal{G}|}{\log |Act(p)|})$.

The related notion $i(\mathcal{G})$ of *intricacy* of a game is defined by:

$$i(\mathcal{G}) = \max_{p \in P} \frac{|Neigh(p)| \times \log |Act(p)|}{\log ||G||}.$$

It is clear that a class of games has small neighborhood iff the intricacy of all games in it is bounded by some constant.

Obviously, bounded neighborhood implies small neighborhood, but not vice-versa. We believe that a very large number of important (classes of) games in economics have the small neighborhood property.

In addition to the quantitative aspect of the size of the neighborhood and/or the action neighborhood, we are also interested in qualitative aspects of mutual strategic influence. To this aim, for a game $\mathcal{G}$ with set P of players, we define the strategic dependency graph as the undirected graph $\mathrm{G}(\mathcal{G}) := \{\{p, q\} | q \in P \wedge p \in Neigh(q)\}$, and the strategic hypergraph $\mathrm{H}(\mathcal{G})$ whose vertices are the players and whose set of hyperedges is $\{\{p\} \cup Neigh(p) | p \in P\}$. We then can immediately define the following classes of structurally restricted games:

Acyclic-Graph Games: Games $\mathcal{G}$ for which $\mathrm{G}(\mathcal{G})$ is acyclic.

Acyclic-Hypergraph Games: Games $\mathcal{G}$ for which $\mathrm{H}(\mathcal{G})$ is acyclic. Note that there are several definitions of hypergraph acyclicity [9]. Here we refer to the broadest (i.e., the most general) one, also known as α-acyclicity [9, 3] (see also Section 2).

Each acyclic graph game is also an acyclic hypergraph game, but not vice-versa. As an extreme example, let $\mathcal{G}$ be a game with player set P in which the utility of each action for each player depends on all other players. Then $\mathrm{G}(\mathcal{G})$ is a clique of size $|P|$ while $\mathrm{H}(\mathcal{G})$ is the trivially acyclic hypergraph having the only hyperedge $\{P\}$.

For strategic games, both the acyclic graph and the acyclic hypergraph assumptions are very severe restrictions, which are rather unlikely to apply in practical contexts. However, there are important generalizations that appear to be much more realistic for practical applications. These concepts are bounded treewidth [38] and bounded hypertree width [16] (see also Section 4), which are suitable measures of the degree of cyclicity of a graph and of a hypergraph, respectively, and can be checked in polynomial time. In particular, each acyclic graph (hypergraph) has treewidth (hypertree width) ≤ 1. It was argued that an impressive number of "real-life" graphs have a very low treewidth [8]. Hypertree width in turn was fruitfully applied in the context of database queries [16] and constraint satisfaction problems [14]. We have, for each constant k, the following restricted classes of games:

Games of treewidth bounded by k: Games $\mathcal{G}$ where the treewidth of $\mathrm{G}(\mathcal{G})$ is $\leq k$.

Games of hypertree width bounded by k: Games $\mathcal{G}$ where the hypertree width of $\mathrm{H}(\mathcal{G})$ is $\leq k$.

In the context of complexity and efficiency studies, it is very important to make clear how an input (in our case, a multiplayer game) is represented. We say that a game is in *general form* if the sets of players and actions are given in extensional form and if the neighborhood and utility functions are polynomially computable functions. Unless otherwise stated, we always assume that games are given in general form. For classes of games having particular properties, some alternative representations have been used by various authors. In the context of games with restricted player interaction, the most used representation is the *graphical normal form (GNF)*. In GNF games, also known as *graphical games* [19, 20, 21, 42], the utility function for each player p is given by a table that displays $p's$ utility as a function of all possible combined strategies of p and p's neighbors, but not of irrelevant other players.

Main results:

- Determining whether a strategic game has a pure Nash equilibrium is NP-complete and remains NP-complete even for following two restricted cases:
  - Games in graphical normal form (GNF) having bounded neighborhood (Theorem 3.1).
  - Acyclic-graph games, and acyclic-hypergraph games (Theorem 3.2).

  The same results hold for Pareto Nash equilibria for pure strategies.
- Determining whether a strategic game has a strong Nash equilibrium is Σ_2^p-complete and thus at the second level of the Polynomial Hierarchy (Theorem 3.5). The proof of this theorem gives us a fresh game-theoretic view of the class Σ_2^P as the class of problems whose positive instances can be solved by a coalition of players who cooperate to provide an equilibrium and win against any other disjoint coalition, which fails in trying to improve the utility for all of its players. In the case of Σ_2 quantified Boolean formulas, the former coalition consists of the existentially quantified variables, and the latter of the universally quantified ones.
- The pure Nash equilibrium existence and computation problems are tractable for games (in whatever representation) that simultaneously have small neighborhood and bounded hypertree width (Theorem 5.3). Observe that each of the two joint restrictions, small neighborhood and bounded hypertree width, is weaker than the restrictions of bounded neighborhood and acyclicity, respectively, of which each by itself does not guarantee tractability. Thus, in order to obtain tractability, instead of strengthening a single restriction, we combined two weaker restrictions. While we think that each of the two strong restrictions is unrealistic, we believe that for many natural games the combined weaker restrictions do apply. In order to prove the tractability result, we establish a relationship between strategic games and the well-known finite domain constraint satisfaction problem (CSP), much studied in the AI and OR literature (see, e.g., [40, 14]). Let us point out that Vickrey and Koller [42] exploited a similar mapping for the different problem of finding Nash equilibria for mixed strategies. However, their approach is different from ours as it represent a game by means of a directed graph rather than by a hypergraph. We prove that each strategic game $\mathcal{G}$ can be translated into a CSP instance of the same hypertree width as $\mathcal{G}$, and whose feasible solutions exactly correspond to the Nash equilibria of the game. We show that this translation is feasible in polynomial time for the classes of games under consideration. We show that, from these results and earlier results on hypertree decompositions of CSPs, it follows that any game $\mathcal{G}$ can be transformed into an equivalent acyclic constraint satisfaction problem (CSP) of size $||\mathcal{G}||^{O(i(\mathcal{G})\times hw(\mathcal{G}))}$, where $hw(G)$ is the hypertree width of the neighborhood hypergraph of $\mathcal{G}$. Acyclic CSPs, in turn, are well-known to be solvable in polynomial time.

	General Players Interaction	Bounded Tree width	Bounded Hypertree Width
General Form	NP-complete	NP-complete	NP-complete
Bounded Neighborhood	NP-complete	PTIME	PTIME
Small Neighborhood	NP-complete	PTIME	PTIME
Graphical Normal Form	NP-complete	LOGCFL-complete	LOGCFL-complete

Figure 1: Complexity of deciding existence of pure Nash equilibria for games in GNF

- By similar techniques we prove that the pure Nash equilibrium existence and computation problems are tractable for games in graphical normal form (GNF) having bounded hypertree width (Theorem 5.3).
- We show that if a strategic game has bounded treewidth, then it also has bounded hypertree width (Theorem 4.2). From this and the last result it follows immediately that the Nash equilibrium existence and computation problems are tractable for games that simultaneously have small neighborhood and bounded treewidth, and for GNF games having bounded treewidth (Corollary 5.5).
- In all above cases where a pure Nash Equilibrium can be computed in polynomial time, also a Pareto Nash equilibrium can be computed in polynomial time (Theorem 5.4).
- In case a game is given in GNF, the problem of determinating a pure Nash equilibrium of a game of bounded hypertree-width (or bounded treewidth) is LOGCFL-complete and thus in the parallel complexity class NC_2 (Theorem 6.1). Membership in LOGCFL follows from the membership of bounded hypertree-width CSPs in LOGCFL [15]. Hardness for LOGCFL is shown by transforming (logspace uniform families of) semi-unbounded circuits of logarithmic depth together with their inputs into strategic games, such that the game admits a Nash equilibrium iff the circuit outputs one on the given input.

Figure 1 summarizes our results on the existence of pure Nash equilibria.

While various authors have dealt with the complexity of Nash equilibria (see, e.g., [12, 35, 23, 24, 6]), most investigations were dedicated to mixed equilibria and — to the best of our knowledge — all complexity results in the present paper are novel. We are not aware of any other work considering the quantitative and structural restrictions on pure games studied here. Note that tree-structured games were considered in [20] in the context of mixed equilibria. It turned out that, for such games, suitable approximation of (mixed) Nash equilibria can be computed in polynomial time. In our future work, we would like to extend our tractability results even to this setting. We are not aware of any work by others on the parallel complexity of equilibria problems. We believe that our present work contributes to the understanding of pure Nash equilibria and proposes appealing and realistic restrictions under which the main computation problems associated to such equilibria are tractable.

2 Games and Nash Equilibria

A *game* $\mathcal{G}$ is a tuple $\langle P, Neigh, Act, U\rangle$, where P is a non-empty set of distinct players and $Neigh$, Act, and U are sets defined as follows. For each player $p \in P$, $Neigh$ contains the set of her neighbors $N_p \subseteq P - \{p\}$, denoted by $Neigh(p)$, Act contains the set of her possible actions, denoted by $Act(p)$, and U contains her utility function $u_p : Act(p) \times_{j \in Neigh(p)} Act(j) \to \Re$.

For a player p, p_a denotes her choice to play the action $a \in Act(p)$. Each possible p_a is called a *strategy* for p, and the set of all strategies for p is denoted by $St(p)$.

For a non-empty set of players $P' \subseteq P$, a *combined strategy* for P' is a set containing exactly one strategy for each player in P'. $St(P')$ denotes the set of all combined strategies for the players in P'. The combined strategy $\mathbf{x}$ is called *global* if $P' = P$. The global strategies are the possible outcomes of the game.

A set of players $K \subseteq P$ is often called a *coalition*. Let $\mathbf{x}$ be a global strategy, K a coalition, and $\mathbf{y}$ a combined strategy for K. Then, we denote by $\mathbf{x}_{-K}[\mathbf{y}]$ the global strategy where, for each player $p \in K$, her individual strategy $x_a \in \mathbf{x}$ is replaced by her individual strategy $x_b \in \mathbf{y}$. If K is a singleton $\{p\}$, we will simply write $\mathbf{x}_{-p}[\mathbf{y}]$.

Let $\mathbf{x}$ be a global strategy, p a player, and u_p the utility function of p. Then, with a small abuse of notation, $u_p(\mathbf{x})$ will denote the output of u_p on the projection of $\mathbf{x}$ to the domain of u_p, i.e., to $St(Neigh(p) \cup \{p\})$.

In the context of complexity and efficiency studies it is very important to make clear how an input (in our case, a multiplayer game) is represented. We say that a game is in *general form* if the sets of players and actions are given in extensional form and if the neighborhood and utility functions are polynomially computable functions. Unless otherwise stated, we always assume that games are given in general form. The following more restrictive classes of (representations of) games have been used by many authors.

Standard Normal Form (SNF): A game with set P of players is in *standard normal form (SNF)* if its utility functions are explicitly represented by a single table or matrix having an entry (or cell) for *each* global strategy $\mathbf{x}$, displaying a list containing for each player p, p's payoff $u_p(\mathbf{x})$ w.r.t. $\mathbf{x}$. (Equivalently, we may describe the utilities by $|P|$ such tables, where the i-th table describes just the payoff of player i.) This is a representation of utility functions often assumed in the literature (see, for instance, Osborne and Rubinstein [32], and Owen [33]). Nonetheless, this is clearly a true restriction because in the general case, even if an utility function is polynomially computable, writing it down in form of a table may require exponential space.

Graphical Normal Form (GNF): A game with set P of players is in *graphical normal form (GNF)* if the utility function for each player p is represented by a separate table containing a cell for each combined strategy $\mathbf{x} \in St(Neigh(p) \cup \{p\})$ of p's set $Neigh(p) \cup \{p\}$ of neighbors, displaying p's payoff $u_p(\mathbf{x})$ w.r.t. $\mathbf{x}$. A representation of utility functions in GNF is illustrated in Example 2.2. The GNF representation has been adopted in several recent papers that study games with a large number of players, where the utility function for each player may depend (directly) only on strategies of a small number of other players [19, 20, 21, 42]. Note that GNF may lead to an exponentially more succinct game representation than both general form and SNF. In the literature, games in GNF are often referred to as *graphical games*. We prefer to use the phrasing *game in graphical normal form*, because this makes clear that we are addressing representational issues.

Let us now formally define the main concepts of equilibria to be further studied in this paper.

Definition 2.1 Let $\mathcal{G} = \langle P, Neigh, A, U\rangle$ be a game and $\mathbf{x}$ be a global strategy for $\mathcal{G}$. Then,

- $\mathbf{x}$ is a pure Nash Equilibrium for $\mathcal{G}$ if, $\forall p \in P$, $\nexists p_a \in St(p)$ such that $u_p(\mathbf{x}) < u_p(\mathbf{x}_{-p}[p_a])$.
- $\mathbf{x}$ is a strong Nash Equilibrium for $\mathcal{G}$ if, $\forall K \subseteq P$, $\forall y \in St(K)$, $\exists p \in K$ such that $u_p(\mathbf{x}) \geq u_p(\mathbf{x}_{-K}[y])$ or, equivalently, if $\forall K \subseteq P$, $\nexists y \in St(K)$ such that, $\forall p \in K$ $u_p(\mathbf{x}) < u_p(\mathbf{x}_{-K}[y])$.
- A pure Nash equilibrium $\mathbf{x}$ is a Pareto Nash Equilibrium for $\mathcal{G}$ if there does not exist a pure Nash equilibrium $\mathbf{y}$ for $\mathcal{G}$ such that, $\forall p \in P$, $u_p(\mathbf{x}) < u_p(\mathbf{y})$.

The sets of pure Nash, strong Nash, and Pareto Nash equilibria of $\mathcal{G}$ are denoted by $\mathcal{NE}(\mathcal{G})$, $\mathcal{SNE}(\mathcal{G})$, and $\mathcal{PNE}(\mathcal{G})$, respectively. It is well known that $\mathcal{SNE}(\mathcal{G}) \subseteq \mathcal{PNE}(\mathcal{G}) \subseteq \mathcal{NE}(\mathcal{G})$.

The existence of a Nash equilibrium does not imply the existence of a strong Nash equilibrium. However, if there exists a Nash equilibrium $\mathbf{x}$, then there exists also a Pareto Nash equilibrium.

F	$P_m R_m$	$P_m R_o$	$P_o R_m$	$P_o R_o$
m	2	2	1	0
o	0	2	1	2

G	$P_m F_m$	$P_m F_o$	$P_o F_m$	$P_o F_o$
m	2	0	0	1
o	2	0	0	1

R	F_m	F_o
m	0	1
o	2	0

P	F_m	F_o
m	2	0
o	0	1

M	R_m	R_o
m	1	0
o	0	2

Figure 2: Utility functions for FRIENDS in GNF

The interaction among players of $\mathcal{G}$ can be naturally represented by a hypergraph $\mathrm{H}(\mathcal{G})$ whose vertices coincide with the players of $\mathcal{G}$ and whose set of (hyper)edges contains for each player p a (hyper)edge $\mathrm{H}(p) = \{p\} \cup Neigh(p)$, referred-to as the *characteristic edge of p*.

A fundamental structural property of hypergraphs is *acyclicity*. Acyclic hypergraphs have been deeply investigated and have many equivalent characterizations (see, e.g., [3]). We recall here that a hypergraph H is acyclic if and only if there is a *join tree* for H, that is, there is a tree $\mathcal{JT}$ whose vertices are the edges of H and, whenever the same player p occurs in two vertices v_1 and v_2, then v_1 and v_2 are connected in $\mathcal{JT}$, and p occurs in each vertex on the unique path linking v_1 and v_2. In other words, the set of vertices in which p occurs induces a (connected) subtree of $\mathcal{JT}$.

Another natural representation of players interaction is through the (undirected) dependency graph $\mathrm{G}(\mathcal{G}) = (P, E)$, whose vertices coincide with the players of $\mathcal{G}$, and $\{p, q\} \in E$ if p is a neighbor of q (and vice versa).

We remark that $\mathrm{G}(\mathcal{G})$ is different from the so called *primal graph PG* of $\mathrm{H}(\mathcal{G})$, which contains an edge for all pairs of vertices that jointly occur in some hyperedge of $\mathrm{H}(G)$. In general, $\mathrm{G}(\mathcal{G})$ is much simpler than *PG*. For instance, consider a game $\mathcal{G}$ with a player p that depends on all other players $q_1, \ldots, q_n$, while these players are independent of each other (but possibly depend on p). Then, $\mathrm{G}(\mathcal{G})$ is a tree. However, the primal graph of $\mathrm{H}(\mathcal{G})$ is a clique with $n + 1$ vertices.

Example 2.2 Let us conclude this section by a motivating example that should sound familiar to everyone, and that we call FRIENDS. It is the game played by a group of persons that have to plan their evening happenings. This game is a generalization of the well known two-person game "battle of sexes". Consider a group of persons consisting of *Giorgio* (short: *G*), *Paola* (*P*), *Francesco* (*F*), *Roberto* (*R*), and *Maria* (*M*). Each of them have to decide to go either to see a *movie* (m) or to see an *opera* (o). However, preferences concern not only the particular option (m or o) to be chosen, but usually also the persons to join for the evening (possibly, depending on the movie or opera choice). For instance, we assume that Francesco is interested in joining Paola and Roberto. He would like to join both of them. However, Paola is an expert of movies and Roberto is an expert of operas. Thus, if it is not possible to go out all together, he prefers to go to the cinema with Paola and to the opera with Roberto. Paola would like to stay with Francesco, and she prefers the movies. Roberto does not like F (perhaps) because he speaks too much and, as we know, he prefers the opera. Also, Maria likes opera and she would like to go there with Roberto. Finally, G is the matchmaker of the group: He has no personal desires, but would like that F and P go together in the same place, better if they go to the cinema. All the utility functions associated to this game are shown in Fig. 2, where we denote the fact that a player X chooses an action a by X_a, e.g., F_m denotes the strategy where F chooses to play the action m.

Here the strategies $\{F_m, P_m, R_o, G_m, M_o\}$, $\{F_m, P_m, R_o, G_o, M_o\}$, $\{F_o, P_o, R_m, G_m, M_m\}$ and $\{F_o, P_o, R_m, G_o, M_m\}$ are Nash equilibria, while the only Pareto Nash equilibria, as well as

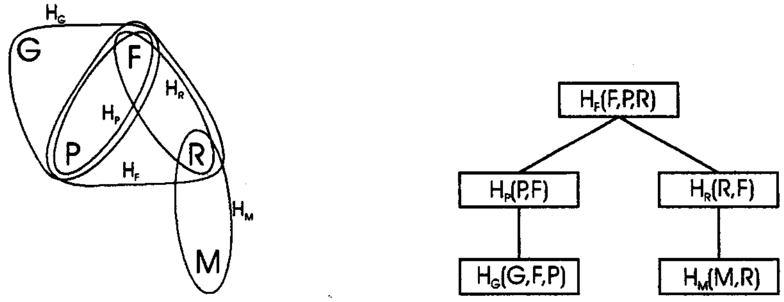

Figure 3: Hypergraph of the FRIENDS game and a join tree for it.

strong Nash equilibria, are $\{F_m, P_m, R_o, G_m, M_o\}$ and $\{F_m, P_m, R_o, G_o, M_o\}$.

Note that the dependency graph associated with the FRIENDS game is not acyclic. For example, the nodes F, P, and G span a cycle in the graph $\mathcal{G}$(FRIENDS). However, the hypergraph H(FRIENDS) associated to the FRIENDS game is acyclic. Figure 2.2 shows this hypergraph and a join tree for it. The notation used for vertices of the join tree makes clear that each vertex of the join tree corresponds to the characteristic edge $\mathrm{H}(p)$ of some player p. □

3 Hard Games

Theorem 3.1 *Deciding whether a game $\mathcal{G}$ has a pure Nash equilibrium is* NP*-complete. Hardness holds even if $\mathcal{G}$ is in GNF, has 3-bounded neighborhood, and where, moreover, each player is allowed to play at most three actions.*

Proof. *Membership.* We can decide that $\mathcal{NE}(\mathcal{G}) \neq \emptyset$ by guessing a global strategy $\mathbf{x}$ and verifying that $\mathbf{x}$ is a Nash equilibrium for $\mathcal{G}$. The latter task can be done in polynomial time. Indeed, for each player p and for each action $a \in Act(p)$, we only have to check that choosing the strategy p_a does not lead to an increment of u_p, and each of these tests is feasible in polynomial time.

Hardness. Recall that deciding whether a Boolean formula in conjunctive normal form $\Phi = c_1 \wedge \ldots \wedge c_m$ over the variables $X_1, \ldots, X_n$ is satisfiable, i.e., deciding whether there exists truth assignments to the variables making each clause c_j true, is an NP-hard problem, even if each clause contains at most three distinct (positive or negated) variables, and each variable occurs in at most three clauses [11]. W.l.o.g, assume Φ contains at least one clause and one variable.

We define a GNF game $\mathcal{G}$ such that: The players are partitioned into two sets P_v and P_c, corresponding to the variables and to the clauses of Φ, respectively; for each player $c \in P_c$, $Neigh(c)$ is the set of the players corresponding to the variables in c, and for each player $v \in P_v$, $Neigh(v)$ is the set of the players corresponding to the clauses in which v occurs; $\{t, f, u\}$ is the set of possible actions for all the players, in which t and f can be interpreted as truth values for variables and clauses. Let $\mathbf{x}$ be a global strategy, then the utility functions are defined as follows. For each player $c \in P_c$, her utility function u_c is such that

- $u_c(\mathbf{x}) = 3$ if (i) c plays t and all of her neighbors play an action in $\{t, f\}$ making the corresponding clause true;
- $u_c(\mathbf{x}) = 2$ if (ii) c plays u and all of her neighbors play an action in $\{t, f\}$ making the corresponding clause false, or (iii) c plays f and there exists $v \in Neigh(c)$ such that v plays u;
- $u_c(\mathbf{x}) = 1$ in all the other cases (iv).

For each other player $v \in P_v$, her utility function u_v is such that

- $u_v(\mathbf{x}) = 3$ if (v) v plays an action in $\{t, f\}$ and all of her neighbors play an action in $\{t, f\}$.
- $u_v(\mathbf{x}) = 2$ if (vi) v plays u and there exists $c \in Neigh(v)$ such that c plays u;
- $u_v(\mathbf{x}) = 1$ in all the other cases (vii).

We claim: Φ is satisfiable $\Leftrightarrow$ $\mathcal{G}$ admits a Nash equilibrium.

($\Rightarrow$) Assume Φ is satisfiable, and take one of such satisfying truth assignments, say σ. Consider the global strategy $\mathbf{x}$ for $\mathcal{G}$ where each player in P_v chooses the action according to σ, and where each player in P_c plays t. Note that, in this case, all players get payoff 3 according to the rules (i) and (v) above, and since 3 is the maximum payoff, $\mathbf{x}$ is a Nash equilibrium for $\mathcal{G}$.

($\Leftarrow$) Let us show that any Nash equilibrium $\mathbf{x}$ for $\mathcal{G}$ corresponds to a satisfying truth assignment of G. We exploit the following properties on the strategies for $\mathcal{G}$.

P_1 : *A strategy* $\mathbf{x}$ *in which a player* $v \in P_v$ *plays* u *cannot be a Nash equilibrium.* In fact, in this case, assuming a Nash equilibrium, first observe that all $c \in Neigh(v)$ must play f for otherwise according to rule (iii) they could improve their payoff. Thus, player v gets payoff 1 (vii), and she can easily increase her payoff to 3 by playing an action in $\{t, f\}$ (v). Contradiction.

P_2 : *A strategy* $\mathbf{x}$ *in which a player* $c \in P_c$ *plays* u *cannot be a Nash equilibrium.* In fact, assume a Nash equilibrium in this case. Then, from (vi), each variable $v \in Neigh(c)$ must play u, for otherwise she could improve her utility. This contradicts P_1.

P_3 : *A strategy in which all players play an action in* $\{t, f\}$ *and the corresponding truth assignment makes a clause* c *false cannot be a Nash equilibrium.* In fact, in this case, assuming a Nash equilibrium, from (ii), c should play u, and this contradicts P_2.

P_4 : *A strategy in which all players play an action in* $\{t, f\}$ *and there exists a player* $c \in P_c$ *that plays* f *cannot be a Nash equilibrium.* In fact, assuming a Nash equilibrium, if the clause corresponding to c is made true by the truth assignments of her neighbors, then c gets payoff 1 (vii), which could be raised to 3 by letting c play t (i). Analogously, if the clause is false, then c gets payoff 1, which could be increases to 2 by letting c play u (ii).

By combining the "$\Rightarrow$"-part of the proof and properties P_1, P_2, P_3 and P_4, we conclude that the only strategies that are Nash equilibria are those corresponding to satisfying truth value assignments to Φ where each player in P_c plays t. Thus, there is a one-to-one correspondence between satisfying truth assignments to the variables of Φ and Nash equilibria of $\mathcal{G}$.

Finally, observe that the tables representing the entries of the utility functions (rules (i)–(vii) above) can be built in polynomial time from Φ. Moreover, due to the structure of Φ and to our construction, each player depends on 3 other players at most. □

We next show that, in the general setting, even if the structure of player interaction is very simple, the problem of deciding whether pure Nash equilibria exist remains hard.

Theorem 3.2 *Deciding whether a (general) game* $\mathcal{G}$ *has a pure Nash equilibrium is* NP*-complete, even if both its dependency graph and its associated hypergraph are acyclic.*

Proof. (Sketch.) After the previous theorem, only hardness remains to be proved. Given a Boolean formula Φ over variables $X_1, ..., X_m$, we define a game $\mathcal{G}$ with m players $X_1, ..., X_m$ corresponding to the variables of Φ, and two additional players T and H.

Any player X_i, $1 \leq i \leq m$, has only two available actions, t and f, corresponding to truth assignments to the corresponding variable of Φ. Then, given a combined strategy $\mathbf{x}$ for $X_1, ..., X_m$, we denote by $\Phi(\mathbf{x})$ the evaluation of Φ on the truth values determined by $\mathbf{x}$. Moreover, the utility function of each player X_i is a constant, say 1. Thus, the choice of X_i is independent of any other player.

Intuitively, the role of player T is to check whether the actions chosen by $X_1, \ldots, X_m$ encode a satisfying truth assignment for Φ. In fact, T may play either s or u, meaning "satisfied" or "unsatisfied," respectively. However, only strategies where T plays s will lead to Nash equilibria, because of her interaction with the player H, whose role is to discard bad strategies, and whose available actions are g and b. Formally, T depends on the players in $\{X_1, \ldots, X_m, H\}$, and her utility function is defined as follows. Let $\mathbf{y} = \mathbf{x}_1 \cup \mathbf{x}_2$ be a combined strategy for $X_1, \ldots, X_m, H, T$, where $\mathbf{x}_1$ encodes a truth-value assignment for Φ and $\mathbf{x}_2$ is a combined strategy for T and H. Then, $u_T(\mathbf{y}) = 1$ if $\Phi(\mathbf{x}_1)$ is true and T plays s, or $\Phi(\mathbf{x}_1)$ is false and $\mathbf{x}_2 = \{T_u, H_g\}$, or $\Phi(\mathbf{x}_1)$ is false and $\mathbf{x}_2 = \{T_s, H_b\}$; otherwise, $u_T(\mathbf{y}) = 0$.

The player H depends only on T and, for any combined strategy $\mathbf{x}$ for H and T, her utility function is the following: $u_H(\mathbf{x}) = 1$ if $\mathbf{x}$ is either $\{T_s, H_g\}$ or $\{T_u, H_b\}$; $u_H(\mathbf{x}) = 0$, otherwise.

Note that, for each satisfying assignment for Φ there is a corresponding Nash equilibrium for $\mathcal{G}$, where $X_1, \ldots, X_m$ play truth values according to this assignment, and T and H play s and g, respectively.

On the other hand, assume that for each combined strategy $\mathbf{x}_1$ for $X_1, \ldots, X_m$ the formula Φ is not satisfied. Then, it is easy to see that, for each combined strategy $\mathbf{x}_2 \in St(\{H, T\})$, $\mathbf{x}_1 \cup \mathbf{x}_2$ is not a Nash equilibrium for $\mathcal{G}$, because either H or T can improve her payoff. Finally observe that the dependency graph $\mathrm{G}(\mathcal{G})$ is a tree, and the hypergraph $\mathrm{H}(\mathcal{G})$ is acyclic. □

The above NP hardness result obviously extends to all considered generalizations of acyclicity.

Let us now draw our attention to Pareto equilibria. It follows trivially from Definition 2.1 that a Pareto Nash equilibrium exists iff a Nash equilibrium exists. We therefore immediately get the following corollary to Theorems 3.1 and 3.2.

Corollary 3.3 *Deciding whether a game $\mathcal{G}$ has a Pareto Nash equilibrium is NP-complete. Hardness holds even if either $\mathcal{G}$ is in graphical normal form and has k-bounded neighborhood, for any fixed constant $k \geq 3$, or if both $\mathrm{G}(\mathcal{G})$ and $\mathrm{H}(\mathcal{G})$ are acyclic.*

However, while checking whether a global strategy $\mathbf{x}$ is a pure Nash equilibrium is tractable, it turns out that checking whether $\mathbf{x}$ is a Pareto Nash equilibrium is a computationally hard task, and is as difficult as checking whether $\mathbf{x}$ is a strong Nash equilibrium.

Theorem 3.4 *Given a game $\mathcal{G}$ and a global strategy $\mathbf{x}$, deciding whether $\mathbf{x} \in \mathcal{PNE}(\mathcal{G})$ (resp., $\mathbf{x} \in \mathcal{SNE}(\mathcal{G})$) is* co-NP*-complete. Hardness holds even if the given strategy $\mathbf{x}$ is a pure Nash equilibrium, and $\mathcal{G}$ is in graphical normal form and has* 3*-bounded neighborhood.*

Proof. (Sketch.) *Membership.* Deciding whether $\mathbf{x} \notin \mathcal{PNE}(\mathcal{G})$ is in NP. First check in polynomial time whether $\mathbf{x} \notin \mathcal{NE}(\mathcal{G})$. If this is not the case, guess a global strategy $\mathbf{y}$ and check in polynomial time whether $\mathbf{y} \in \mathcal{NE}(\mathcal{G})$ and $\mathbf{y}$ dominates $\mathbf{x}$.

Hardness. Deciding whether a Boolean formula Φ is not satisfiable is co-NP-complete; the hardness holds even if each clause contains three variables at most, and if each variable occurs in three clauses at most. We define a GNF game $\mathcal{G}$ in the same way as in the proof of Theorem 3.1, but for the following modifications. Each player may choose an additional action w getting payoff 2, no matter what the other players do. Rules (i),…,(vii) have an additional constraint stating that they are applicable only if all the adjacent players do not play w. Finally, we add the rule (viii), which says that $u_p(\mathbf{x}) = 0$ if p does not play w and there exists an adjacent player choosing w. Moreover, w.l.o.g., we assume that the formula Φ is such that the graph $\mathrm{G}(\mathcal{G})$ is connected.

Then, it can be seen that $\mathcal{G}$ has a Nash equilibrium for each satisfying truth assignment of Φ (if any), plus a Nash equilibrium $\mathbf{x}_w$ where each player chooses w and thus gets payoff 2. Since in the strategies corresponding to satisfying truth assignments each player gets payoff 3, it follows that $\mathbf{x}_w \in \mathcal{PNE}(\mathcal{G})$ if and only if Φ is not satisfiable.

The same relationship holds for strong Nash equilibria, that is, $\mathbf{x}_w \in \mathcal{SNE}(\mathcal{G})$ if and only if Φ is not satisfiable. Note that, if there is a satisfying truth assignment for Φ, then $x_w \notin \mathcal{PNE}(\mathcal{G})$, and hence $\mathbf{x}_w \notin \mathcal{SNE}(\mathcal{G})$, too. Thus, it remains to prove that if there are no satisfying assignment for Φ then $\mathbf{x}_w \in \mathcal{SNE}(\mathcal{G})$. Assume by contradiction that $\mathbf{x}_w \notin \mathcal{SNE}(\mathcal{G})$, let K be any coalition of players, and $\mathbf{y} \in St(K)$ a combined strategy for K. The only way for improving the payoff for every player in K is that all the variables in K plays an action in $\{t, f\}$, and the clauses play t.

However, since $\mathrm{G}(\mathcal{G})$ is connected, if K does not contain all vertices of G, there is some player $q' \notin K$, and thus playing w, such that $\{q', p\}$ is an edge of $\mathrm{G}(\mathcal{G})$, for some $p \in K$. From rule (viii), this entails that $u_p(\mathbf{x}_{-K}[\mathbf{y}]) = 0$, that is a player of the coalition gets a lower payoff. It follows that every good coalition involves all players, that is, all vertices of the graph, and encodes a satisfying truth assignment. Contradiction. □

Deciding whether a game has a strong Nash equilibrium turns out to be much more difficult than deciding the existence of pure (Pareto) Nash equilibria. Indeed, this problem is located at the second level of the polynomial hierarchy.

Theorem 3.5 *Given a game $\mathcal{G}$ in GNF deciding whether $\mathcal{G}$ has strong Nash equilibria is Σ_2^P-complete.*

Proof. (Sketch.) *Membership.* We can decide that $\mathcal{SNE}(\mathcal{G}) \neq \emptyset$ by guessing a global strategy $\mathbf{x}$ and verifying that $\mathbf{x}$ is a strong Nash equilibrium for $\mathcal{G}$. From Theorem 3.4, this latter task is feasible in co-NP. It follows that the problem is in Σ_2^P.

Hardness. Let $F = \exists\alpha_1, \ldots \alpha_n \forall\beta_1, \ldots \beta_m \Phi$ be a quantified Boolean formula, where Φ is a Boolean formula over the propositional variables $\alpha_1, \ldots \alpha_n, \beta_1, \ldots \beta_m$. W.l.o.g., assume that $n \geq 2$. Deciding the validity of such formulas is a well-known Σ_2^P-complete problem. We define a game $\mathcal{G}$ as follows. The players P of $\mathcal{G}$ coincide with the variables of F and are partitioned in two sets P_α and P_β corresponding to existentially and universally quantified variables, and thus called existential and universal players, respectively. The set of actions of every player includes two actions t and f, that we can interpret as truth values assigned to the corresponding variables occurring in Φ. Thus, a global strategy $\mathbf{x}$ where all players play such truth value actions corresponds to a truth value assignment for Φ whose outcome is denoted by $\Phi(\mathbf{x})$.

Let β be a universal player in P_β. Her set of actions is $Act(\beta) = \{t, f\}$, and her utility function u_β is defined as follows. For any strategy $\mathbf{x}$, $u_\beta(\mathbf{x}) = 1$ if all variables play truth values and $\Phi(\mathbf{x}) = \mathit{false}$; $u_\beta(\mathbf{x}) = 0$, otherwise. Thus, universal players have their best payoff when they are able to make the formula *false*. The opposite happens for existential players that try to satisfy Φ.

For any existential player $\alpha \in P_\alpha$, the set of actions is $Act(\alpha) = \{t, f, u, s\}$. Moreover, for any global strategy $\mathbf{x}$, her utility function is the following:

- $u_\alpha(\mathbf{x}) = 0$ if (i) all players play truth values and $\Phi(\mathbf{x}) = \mathit{false}$, or (ii) α plays a truth value, somebody plays s, and nobody plays u, or (iii) α plays u and some player in P_α does not play s;
- $u_\alpha(\mathbf{x}) = 1$ if (iv) α plays s (no matter what other players do);
- $u_\alpha(\mathbf{x}) = 2$ if (v) either all players play truth values and Φ is satisfied by $\mathbf{x}$, or α plays a truth value and some player plays u;
- $u_\alpha(\mathbf{x}) = 3$ if (vi) α plays u and all existential players play s.

In the full paper [13] we show that the Nash equilibria of $\mathcal{G}$ are in a one-to-one correspondence to the satisfying truth assignments of Φ and prove that the quantified formula F is valid if and only if $\mathcal{G}$ has strong Nash equilibria. □

4 Treewidth and Hypertree Width of Games

We assume the reader is familiar with the notion of treewidth [38], a well-known measure of graph cyclicity. *Hypertree width* provides a measure of the degree of cyclicity of hypergraphs.

Let $\mathrm{H} = (V, E)$ be a hypergraph. Denote by $vert(\mathrm{H})$ and $edges(\mathrm{H})$ the sets V and E, respectively. Moreover, for any set of edges $E' \subseteq edges(\mathrm{H})$, let $vert(E') = \bigcup_{h \in E'} h$.

A *hypertree for a hypergraph* H is a triple $\langle T, \chi, \lambda \rangle$, where $T = (N, E)$ is a rooted tree, and χ and λ are labelling functions which associate to each vertex $p \in N$ two sets $\chi(p) \subseteq vert(\mathrm{H})$ and $\lambda(p) \subseteq edges(\mathrm{H})$. If $T' = (N', E')$ is a subtree of T, we define $\chi(T') = \bigcup_{v \in N'} \chi(v)$. We denote the root of T by $root(T)$. Moreover, for any $p \in N$, T_p denotes the subtree of T rooted at p.

Definition 4.1 ([16]) A *hypertree decomposition* of a hypergraph H is a hypertree $HD = \langle T, \chi, \lambda \rangle$ for H, where $T = (N, E)$, which satisfies all the following conditions:

1. for each edge $h \in edges(\mathrm{H})$, there exists $p \in N$ such that $vert(h) \subseteq \chi(p)$ (we say that p *covers* h);
2. for each vertex $Y \in vert(\mathrm{H})$, the set $\{p \in N \mid Y \in \chi(p)\}$ induces a (connected) subtree of T;
3. for each $p \in N$, $\chi(p) \subseteq vert(\lambda(p))$;
4. for each $p \in N$, $vert(\lambda(p)) \cap \chi(T_p) \subseteq \chi(p)$.

The *width* of a hypertree decomposition $\langle T, \chi, \lambda \rangle$ is $max_{p \in vertices(T)} |\lambda(p)|$. The *hypertree width* $hw(\mathrm{H})$ of H is the minimum width over all its hypertree decompositions.

Formally, a class of games C is said to have bounded hypertree-width (resp., treewidth) if there is a finite k such that, for each game $\mathcal{G} \in C$, $\mathcal{G}$ has k-bounded hypertree width (resp., treewidth).

Note that for any constant k checking whether a graph has treewidth at most k or a hypergraph has hypertree-width at most k is feasible in polynomial time ([4] and [16], respectively).

Theorem 4.2 *For each game $\mathcal{G}$, $hypertreewidth(\mathrm{H}(\mathcal{G})) \leq treewidth(\mathrm{G}(\mathcal{G}))$.*

Proof. (Sketch.) Let *TD* be a tree decomposition of $\mathrm{G}(\mathcal{G})$ and let k be its width. Then, we show that there is a hypertree decomposition of $\mathrm{H}(\mathcal{G})$ having width k. Recall that $\mathrm{H}(\mathcal{G})$ contains, for each player p, the characteristic edge $\mathrm{H}(p)$. Let $HD = \langle T, \chi, \lambda \rangle$ be a hypertree whose tree T is isomorphic to the tree decomposition. Let δ be a function encoding the one-to-one correspondence between vertices of T and vertices of *TD*. Then, for each vertex $v \in T$, $\lambda(v) = \{\mathrm{H}(p) \mid p \in \delta(v)\}$, i.e., it contains the characteristic edge of the players occurring in the vertex of the tree decomposition corresponding to v. Moreover, $\chi(v)$ is simply the set of all vertices occurring in the edges in $\lambda(v)$. Note that the width of *HD* is k, and it is easy to see that Conditions 1, 3, and 4 of Definition 4.1 hold for *HD*. The proof that Condition 2 holds is quite involved, and is given in the full paper [13]. □

Theorem 4.3 *There are classes C of games having hypertree width 1 but unbounded treewidth.*

Proof. (Sketch.) Take the class of all games where every player depends on all other players. For every such a game $\mathcal{G}$, $\mathrm{H}(\mathcal{G})$ is acyclic and thus its hypertree width is 1, while $\mathrm{G}(\mathcal{G})$ is a clique containing all players and its treewidth is the number of players minus 1. □

5 Easy Games

Before we deal with tractable games in GNF, let us establish that all computational problems dealt with in this paper are tractable for games in SNF. Actually, they can be carried out in logarithmic space and are thus in a very low complexity class that contains highly parallelizable problems only.

Theorem 5.1 *Given a game in standard normal form, the following tasks are all feasible in logarithmic space: Determining the existence of a pure Nash equilibrium, a pure Pareto equilibrium, or a strong Nash equilibrium, and computing all such equilibria.*

Proof. (Sketch.) Let P, as usual, denote the set of players. We assume w.l.o.g. that each player has at least two possible actions (in fact, a player with a single action can be eliminated from the game by a simple logspace transformation, yielding an equivalent game). The size of the input matrix is thus at least $2^{|P|}$.

Given that *all* possible global strategies are explicitly represented, each corresponding to a table cell (which can be indexed in logarithmic space), the Nash equilibria are easily identified by scanning all global strategies $\mathbf{x}$ keeping a logspace index of $\mathbf{x}$ in the worktape, and checking in logarithmic space whether no player can improve her utility by choosing another action.

The Pareto equilibria can be identified by successively enumerating all Nash equilibria ($\mathbf{x}$), and by an additional loop for each $\mathbf{x}$, enumerating all Nash equilibria $\mathbf{y}$ (indexed in logarithmic space as above) and outputting $\mathbf{x}$ if there is no $\mathbf{y}$, with $\forall p \in P$, $u_p(\mathbf{x}) > u_p(\mathbf{y})$. The latter condition can be tested by means of a simple scan of the players.

Strong Nash equilibria can be identified by enumerating all Nash equilibria and by scanning all the coalitions of the players (which can be indexed in logarithmic space, as their number is bounded by $2^{|P|}$) in order to discard those equilibria $\mathbf{x}$ for which there exists a coalition $K \subseteq P$ and a combined strategy y for K, such that for each $p \in K$, $u_p(\mathbf{x}) < u_p(\mathbf{x}_{-K}[y])$. Finally, note that, for a fixed coalition K, the enumeration of all the combined strategies y for K can be carried out by means of an additional nested loop, requiring logarithmic space for indexing each such strategy. □

Let us now consider games in GNF. We first establish an interesting connection between constraint satisfaction problems and games. An instance of a *constraint satisfaction problem (CSP)* (also *constraint network*) is a triple $I = (Var, U, \mathcal{C})$, where Var is a finite set of variables, U is a finite domain of values, and $\mathcal{C} = \{C_1, C_2, \ldots, C_q\}$ is a finite set of constraints. Each constraint C_i is a pair (S_i, r_i), where S_i is a list of variables of length m_i called the *constraint scope*, and r_i is an m_i-ary relation over U, called the *constraint relation*. (The tuples of r_i indicate the allowed combinations of simultaneous values for the variables S_i). A *solution* of a CSP instance is a substitution $\theta : Var \longrightarrow U$, such that for each $1 \leq i \leq q$, $S_i\theta \in r_i$. Since we are interested in CSPs associated to games, where variables are players of games, we will use interchangeably the terms variable and player, whenever no confusion arises.

Let $\mathcal{G} = \langle P, Neigh, A, U\rangle$ be a game and $p \in P$ a player. Define the *Nash constraint* $NC(p) = (S_p, r_p)$ as follows: The scope S_p consists of the players in $\{p\} \cup Neigh(p)$, and the relation r_p contains precisely all combined strategies $\mathbf{x}$ for $\{p\} \cup Neigh(p)$ such that there is no $y_p \in St(p)$ such that $u_p(\mathbf{x}) < u_p(\mathbf{x}_{-p}[y_p])$. Note that, for each Nash equilibrium $\mathbf{x}$ of $\mathcal{G}$, $\mathbf{x} \cap St(S_p)$ is in r_p.

The constraint satisfaction problem associated with $\mathcal{G}$, denoted by $CSP(\mathcal{G})$, is the triple $(Var, U, \mathcal{C})$, where $Var = P$, the domain U contains all the possible actions of all players, and $\mathcal{C} = \{NC(p) \mid p \in P\}$, i.e., it is the set of Nash constraints for the players in $\mathcal{G}$.

Theorem 5.2 *A strategy $\mathbf{x}$ is a pure Nash equilibrium for a game $\mathcal{G}$ if and only if it is a solution of $CSP(\mathcal{G})$.*

Proof. (Sketch.) Let $\mathbf{x}$ be a Nash equilibrium for $\mathcal{G}$ and let p any player. Then, for each strategy $p_a \in St(p)$, $u_p(\mathbf{x}) \geq u_p(\mathbf{x}_{-p}[p_a])$. Since u_p depends only on the players in $\{p\} \cup Neigh(p)$, their combined strategy $\mathbf{x}' \subseteq \mathbf{x}$ is a tuple of $NC(p)$, by construction. It follows that the substitution assigning to each player p its individual strategy $p_a \in \mathbf{x}$ is a solution of $CSP(\mathcal{G})$.

On the other hand, consider any solution θ of $CSP(\mathcal{G})$, and let p be any player. Let $P' = \{p\} \cup Neigh(p)$ and $\mathbf{x}'$ the combined strategy $\{\theta(q) \mid q \in P'\}$. Then, $\mathbf{x}'$ is a tuple of $NC(p)$, because θ is a solution of $CSP(\mathcal{G})$. Thus, for each p, by definition of $NC(p)$, there is no individual strategy for p that can increase her utility, given the strategies of the other players. It follows that the global strategy containing $\theta(p)$ for each player p is a Nash equilibrium for $\mathcal{G}$. □

Theorem 5.3 *Deciding the existence of pure Nash equilibria, as well as computing a Nash equilibrium is feasible in polynomial time for all classes $\mathcal{C}$ of games having bounded hypertree-width and such that every game $\mathcal{G} \in \mathcal{C}$ has small neighborhood or is in graphical normal form. Moreover, all pure Nash equilibria of such games can be computed in time polynomial in the combined size of input and output.*

Proof. (Sketch.) Let $\mathcal{G} = \langle P, Neigh, A, U\rangle$ be a game in a class $\mathcal{C}$ of games having small neighborhood, and hence bounded intricacy. We show that $NC(p) = (S_p, r_p)$ can be computed in polynomial time. We initialize r_p with all the combined strategies for $\{p\} \cup Neigh(p)$.

By definition of intricacy, $|Neigh(p)| \leq i(\mathcal{G}) \log(||\mathcal{G}||)/\log(|Act(p)|)$ holds. Therefore, it is easy to see that $|Act(p)|^{|Neigh(p)|} \leq ||\mathcal{G}||^{i(\mathcal{G})}$. Moreover, $\mathcal{C}$ has bounded intricacy, and thus for some constant t, for each game in $\mathcal{C}$, $i(\mathcal{G}) \leq t$. It follows that the initialization step takes polynomial time (in the size of $\mathcal{G}$).

Then, for each tuple $\mathbf{x}$ in r_p we have to check whether it should be kept in r_p or not. Let $m = u_p(\mathbf{x})$. For each action $a \in Act(p)$, compute in polynomial time $m' = u_p(\mathbf{x}_{-p}[p_a])$ and delete $\mathbf{x}$ if $m' > m$.

It follows that $CSP(\mathcal{G})$ can be computed in polynomial time from $\mathcal{G}$. Furthermore, since $\mathcal{G}$ has k-bounded hypertree width, for some fixed k, and the hypergraph H associated to $CSP(\mathcal{G})$ is the same as H$(\mathcal{G})$, we can compute in polynomial time a hypertree decomposition of H having width at most k, and eventually solve $CSP(\mathcal{G})$ by using the techniques described in [15].

A similar reasoning applies if $\mathcal{G}$ is in GNF. In this case, the utility functions are explicitly given in input in a tabular form, and thus the computation of Nash constraints is easier. In fact, this task is feasible in logspace for GNF games. □

In the full paper [13] we show similar results for Pareto equilibria:

Theorem 5.4 *Deciding the existence of Pareto Nash equilibria, as well as computing a Pareto Nash equilibrium for pure strategies is feasible in polynomial time for all classes $\mathcal{C}$ of games having bounded hypertree-width and such that every game $\mathcal{G} \in \mathcal{C}$ has small neighborhood or is in graphical normal form.*

From the above theorems and Theorem 4.2, we immediately get tractability results for bounded treewidth games.

Corollary 5.5 *Deciding the existence of pure (Pareto) Nash equilibria, as well as computing a pure (Pareto) Nash equilibrium is feasible in polynomial time for all classes $\mathcal{C}$ of games having bounded treewidth and such that every game $\mathcal{G} \in \mathcal{C}$ has small neighborhood or is in graphical normal form. Moreover, all Nash equilibria of such games can be computed in time polynomial in the combined size of input and output.*

6 Parallel Complexity of Easy Games

The complexity class LOGCFL consists of all decision problems that are logspace reducible to a context-free language. We have: $AC_0 \subseteq NC_1 \subseteq$ LOGSPACE $\subseteq$ NL $\subseteq$ LOGCFL $\subseteq AC_1 \subseteq NC_2 \subseteq$ P (see [18, 36]). All problems in LOGCFL are highly parallelizable.

Theorem 6.1 *The existence problem for pure Nash equilibria is* LOGCFL*-complete for the following classes of strategic games in graphical normal form: acyclic-graph games, acyclic-hypergraph games, games of bounded treewidth, and games of bounded hypertree-width.*

Proof. (Idea.) Membership in LOGCFL follows from the fact that our translation into CSP is width-preserving and feasible in LOGSPACE, and the fact that bounded-width CSPs are solvable in LOGCFL. For proving hardness, we exploit a result in [41] stating that LOGCFL coincides with the class SAC^1 of problems solvable by logspace-uniform families of semi-unbounded AC_1 Boolean circuits (SAC^1 circuits) of depth $O(\log n)$. In such circuits the OR gates have unbounded fan-in while all AND gates have fan-in ≤ 2, and $\neg$-gates are at the bottom level only. In the full paper [13], we show that the evaluation problem of SAC^1 circuits can be transformed in logspace into the considered Nash equilibrium existence problems. □

Corollary 6.2 *For the classes of games mentioned in Theorem 6.1, the computation of a single pure Nash equilibria can be done in functional LOGCFL, and is therefore in the parallel complexity class* NC_2.

Acknowledgment. Georg Gottlob's work was supported by the Austrian Science Fund (FWF) under project No. Z29-N04 and by the GAMES Network of Excellence of the EU. We thank the anonymous referees and Tuomas Sandholm for useful comments.

References

[1] R.J. Aumann, Accetable points in general cooperative n-person games. *Contribution to the Theory of Games*, volume IV, Princeton University Press, 1959.

[2] R.J. Aumann, What is Game Theory Trying to Accomplish? in *Frontiers of Economics*, pp. 28–76, Okford: Basil Blackwell, 1985.

[3] C. Beeri, R. Fagin, D. Maier, and M. Yannakakis. On the Desiderability of Acyclic Database Schemes. *Journal of the ACM*, 30(3):479–513, 1983.

[4] H.L. Bodlaender. Treewidth: Algorithmic Techniques and Results. *In Proc. of MFCS'97*, Bratislava. LNCS 1295, Springer, pp. 19-36, 1997.

[5] A. Borodin, S.A. Cook, P.W. Dymond, W.L. Ruzzo and M. Tompa, Two applications of inductive counting for complementation problems, *SIAM Journal on Computing*, v.18 n.3, pp.559-578, June 1989.

[6] V. Conitzer and T. Sandholm. Complexity Results about Nash Equilibria. *to appear in Proc. IJCAI'03.*

[7] X. Deng, C.H. Papadimitriou, and S. Safra, On the complexity of equilibria, *in Proc. of the 34th Annual ACM Symposium on Theory of Computing (STOC 2002)*, pp. 67-71, ACM, Montrial, Canada, 2002.

[8] R.G. Downey and M.R. Fellows, Fixed-Parameter Tractability and Completeness I: Basic Results, *SIAM Journal on Computing*, 24(4), pp. 873-921, 1995.

[9] R. Fagin. Degrees of acyclicity for hypergraphs and relational database schemes. *Journal of the ACM*, 30(3):514–550, 1983.

[10] D. Fotakis, S.C. Kontogiannis, E. Koutsoupias, M. Mavronicolas and P.G. Spirakis, The Structure and Complexity of Nash Equilibria for a Selfish Routing Game, *in Proc. of the 29th International Colloquium on Automata, Languages and Programming (ICALP 2002)*, pp. 123-134, Springer, Malaga, Spain, 2002.

[11] M.R. Garey and D.S.Johnson, *Computers and Intractability. A Guide to the Theory of*NP*-completeness*, Freeman and Comp., NY, USA, 1979.

[12] I. Gilboa and E. Zemel, Nash and correlated equilibria: Some complexity consideration, *Games and Economic Behaviour*, 1, pp. 80-93, 1989.

[13] G. Gottlob, G. Greco, and F. Scarcello. Pure Nash Equilibria: Hard and Easy Games. (Full version.) *To appear.* Draft in progress available at:
`www.dbai.tuwien.ac.at/staff/gottlob/purenash.ps`

[14] G. Gottlob, N. Leone, and F. Scarcello. A Comparison of Structural CSP Decomposition Methods, *Artificial Intelligence*, 124(2), pp. 243-282, 2000.
[15] G. Gottlob, N. Leone, F. Scarcello, The complexity of acyclic conjunctive queries, *Journal of the ACM*, 48(3), pp. 431-498, 2001.
[16] G. Gottlob, N. Leone , F. Scarcello, Hypertree decompositions and tractable queries, *Journal of Computer and System Sciences*, 64(3), pp. 579-627, 2002.
[17] G. Gottlob, N. Leone, F. Scarcello, Computing LOGCFL certificates, *Theoretical Computer Science*, 270(1-2), pp. 761-777, 2002.
[18] D.S. Johnson, A Catalog of Complexity Classes, *Handbook of Theoretical Computer Science, Volume A: Algorithms and Complexity*, pp. 67-161, 1990.
[19] M. Kearns, M.L. Littman and S. Singh, An Efficient Exact Algorithm for Singly Connected Graphical Games, *in Proc. of the Intl. conf. on Neural Information Processing Systems (NIPS01)*, 2001.
[20] M. Kearns, M.L. Littman and S. Singh, Graphical Models for Game Theory,Graphical Models for Game Theory. *in Proc. of the International Conference on Uncertainty in AI (UAI01)*, 2001.
[21] M. Kearns and Y. Mansour, Efficient Nash Computation in Large Population Games with Bounded Influence, *in Proc. of the International Conference on Uncertainty in AI (UAI02)*, 2002.
[22] M. Kearns and L. Ortiz, Nash Propagation for Loopy Graphical Games, *in Proc. of the International conference on Neural Information Processing Systems (NIPS02)*, 2002.
[23] D. Koller and N. Megiddo, The complexity of two-person zero-sum games in extensive form, *Games and Economic Behavior*, 4, pp. 528-552, 1992.
[24] D. Koller and N. Megiddo, Finding mixed strategies with small supports in extensive form games, *International Journal of Game Theory*, 25, pp. 73-92, 1996.
[25] D. Koller, N. Megiddo and B. von Stengel, Efficient computation of equilibria for extensive two-person games, *Games and Economic Behavior*, 14, pp. 220-246, 1996.
[26] E. Koutsoupias and C.H. Papadimitriou, Worst Case Equilibria, *In Proc. of the 16th Symposium on Theoretical Aspects of Computer Science*, pp. 404-413, 1999.
[27] E. Maskin, The Theory of Implementation in Nash Equilibrium, *Social Goals and Organization: Essays in memory of Elisha Pazner*, pp. 173-204, Cambridge University Press, 1985.
[28] R.D. McKelvey and A. McLennan, Computation of equilibria in finite Games, in *Handbook of Computational Economics*, H. Amman, D. Kendrick and J. Rust eds, pp. 87-142, 1996.
[29] N. Megiddo and C.H. Papadimitriou, On Total Functions, Existence Theorems, and Computational complexity, *Theoretical Computer Science*, 81(2), pp. 317-324, 1991.
[30] D. Monderer and L.S. Shapley, Potential games, *Games and Economic Behavior*, 1993.
[31] J.F. Nash, Non-cooperative Games, *Annals of Mathematics*, 54(2), pp. 286-295, 1951.
[32] M.J. Osborne and A. Rubinstein, *A Course in Game Theory*, MIT Presss, 1994.
[33] G. Owen, *Game Theory*, Academic Press, New York, 1982.
[34] C.H. Papadimitriou and M. Yannakakis, On the Complexity as Bounded Rationality, *In Proc. of the 26th Annual ACM Symposium on Theory of Computing*, pp. 726-733, 1994.
[35] C.H. Papadimitriou, On the Complexity of the Parity Argument and Other Inefficient Proofs of Existence, *Journal of Computer and system Sciences*, 48(3), pp. 498-532, 1994.
[36] C.H. Papadimitriou, *Computational Complexity*, Addison-Wesley, Reading, Mass, 1994.
[37] C.H. Papadimitriou, Algorithms, Games, and the Internet, *In Proc. of the 28th International Colloqium on Automata, Languages and Programming*, pp. 1-3, 2001.
[38] N. Robertson and P.D.Seymour, Graph Minors II. Algorithmic aspects of tree width, *Journal of Algorithms*, 7, pp. 309-322, 1986.
[39] R. Rosenthal, A Class of Games possessing pure-strategy Nash equilibria, *International Journal of Game Theory*, 2, pp. 65-67, 1973.
[40] M.Y. Vardi, Constraint Satisfaction and Database Theory: a Tutorial, *In Proc. of the 19-th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pp. 76-85, Dallas, Texas, USA, 2000.
[41] H. Venkateswaran, Properties that characterize LOGCFL, *Journal of Computer and System Sciences*, v.43 n.2, pp.380-404, 1991.
[42] D. Vickrey and D. Koller. Multi-Agent Algortihms for Solving Graphical Games. *In Proc. of the 18-th National Conference on Artificial Intelligence*, pp.345-351, Edmonton, Alberta, Canada, 2002.