Theoretical Computer Science 410 (2009) 3901–3924

Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

# On the complexity of constrained Nash equilibria in graphical games[☆]

Gianluigi Greco[a], Francesco Scarcello[b,*]

[a] *Dipartimento di Matematica, Università della Calabria, Rende, Italy*
[b] *DEIS, Università della Calabria, Rende, Italy*

ARTICLE INFO

*Article history:*
Received 1 August 2008
Received in revised form 27 February 2009
Accepted 24 May 2009
Communicated by P. Spirakis

*Keywords:*
Game theory
Graphical games
Nash equilibria
Computational complexity
Treewidth

ABSTRACT

A widely accepted rational behavior for non-cooperative players is based on the notion of Nash equilibrium. Although the existence of a Nash equilibrium is guaranteed in the mixed framework (i.e., when players select their actions in a randomized manner) in many real-world applications the existence of "any" equilibrium is not enough. Rather, it is often desirable to single out equilibria satisfying some additional requirements (in order, for instance, to guarantee a minimum payoff to certain players), which we call *constrained Nash equilibria*.

In this paper, a formal framework for specifying these kinds of requirement is introduced and investigated in the context of graphical games, where a player $p$ may directly be interested in some of the other players only, called the neighbors of $p$. This setting is very useful for modeling large population games, where typically each player does not directly depend on all the players, and representing her utility function extensively is either inconvenient or infeasible.

Based on this framework, the complexity of deciding the existence and of computing constrained equilibria is then investigated, in the light of evidencing how the intrinsic difficulty of these tasks is affected by the requirements prescribed at the equilibrium and by the structure of players' interactions. The analysis is carried out for the setting of mixed strategies as well as for the setting of pure strategies, i.e., when players are forced to deterministically choose the action to perform. In particular, for this latter case, restrictions on players' interactions and on constraints are identified, that make the computation of Nash equilibria an easy problem, for which polynomial and highly-parallelizable algorithms are presented.

© 2009 Elsevier B.V. All rights reserved.

## 1. Introduction

### 1.1. Graphical games and constrained equilibria

**Graphical games.** Game theory is a mathematical framework for dealing with interactions among autonomous rational agents (see, e.g., [46,45]). A (strategic) *game* consists of a set $P$ of players, each one having to decide the most convenient

☆ A preliminary version of part of this paper appeared in the proceedings of ECAI'04 [G. Greco, S. Scarcello, Constrained pure Nash equilibria in graphical games, in: Proc. of the 16th Eureopean Conference on Artificial Intelligence, ECAI'04, Valencia, Spain, 2004, pp. 181–185] and UAI'05 [G. Greco, F. Scarcello, Bounding the uncertainty of graphical games: The complexity of simple requirements, pareto and strong Nash equilibria, in: Proc. of the 21st Conference on Uncertainty in Artificial Intelligence, UAI'05, 2005, pp. 225–232].

* Corresponding author. Tel.: +39 0984 494752; fax: +39 0984 494713.
*E-mail addresses:* ggreco@mat.unical.it (G. Greco), scarcello@deis.unical.it (F. Scarcello).

0304-3975/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2009.05.030

action (also called *strategy*) to play. When a strategy is chosen for all players, possibly according to some probability distribution, each one gets a payoff that is determined by her choice, as well as by the choices of the other players involved in the game. In the non-cooperative setting we shall consider in this paper, the aim of each player is to maximize her own payoff.

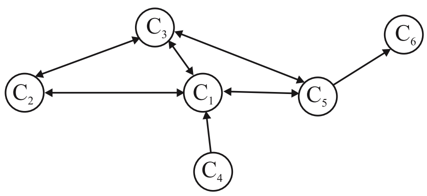
Classically, strategic games are assumed to be given in the so-called *normal form*, i.e., by means of tables where the payoffs of each player $p$ are represented through a table entry for *each* combination of players' strategies (see, e.g., [46,45,57]). Note that this approach is clearly infeasible for games involving a large number of players, such as groups of agents interacting over the internet. For instance, a strategic game played by $n$ players, each one having two available actions, is described by $n$ matrices each of size $2^n$ (or by a single table having $2^n$ cells with $n$ entries per cell). Hence, it comes with no surprise that classes of games with succinct representations attracted much research. In particular, two main approaches have been pursued in the literature, namely:

(1) to exploit context-specific restrictions of agents' utility functions, such as symmetries over the agents, or utilities depending on resources hold by the agents—noticeable examples are the classes of *symmetric games*, *congestion games* [51] (or, *exact potential games* [41]), *local-effect games* [38], *action-graph games* [4], and *bounded influence games* [35]; and,
(2) to represent direct interactions among the agents only, as to take advantage of those scenarios where players' payoffs are affected by decisions of a possibly "small" number of other players. In fact, this perspective characterizes various proposals for describing *influence diagrams* in decision theory, such as the *multi-agent influence diagrams* [56], the *networks of influence diagrams* [22], and the *game networks* [42], just to cite a few.

In this paper, the latter approach to compactly specify strategic games is considered, by focusing on one of the most interesting and well-studied classes therein, that is, the class of *graphical games*, firstly formalized by [34].

In this representation, games are described by means of graphs whose nodes are the players and where the payoffs of each player $p$ are defined in terms of the strategy played by $p$ and of the strategies played by those players $p$ is directly interested in, called the *neighbors of p* and denoted by $Neigh(p)$. In particular, payoffs of $p$ are represented by means of a table having exactly an entry for each combination of strategies for players in $\{p\} \cup Neigh(p)$. Thus, the graphical representation of a game is very compact, if compared to its normal form. This is illustrated in the following example.

**Example 1.1.** Consider a set $\{C_1, C_2, C_3, C_4, C_5, C_6\}$ of companies in a market modelled by means of the game $\mathcal{G}_c$. Each company has often a limited number of other market players (directly) influencing its strategic decisions. These relevant players are usually known and constitute the neighborhood of the company. For instance, assume that: $C_1$, $C_2$ and $C_3$ mutually influence each other; $C_1$, $C_3$, and $C_5$ mutually influence each other; $C_5$ influences $C_6$; and, $C_4$ influences $C_1$. The neighborhood relationship $\mathcal{G}_c$ is depicted below.




Assuming that each company has two strategies at most, the normal form representation would require 6 matrices of size $2^6$ each, while the graphical representation requires 6 tables ranging from $2^2$ to $2^5$ entries (for the table of $C_1$). Note that, despite the succinct representation, 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. For instance $C_6$ is well influenced by decisions of $C_4$ (e.g., through the strategies chosen by $C_5$ and $C_1$). $\lhd$

**Constrained Nash equilibria.** A widely accepted formalization of the rational behavior for a set of interacting players is the notion of Nash equilibrium, originally introduced by Nash [43]. This notion is aimed at singling out those outcomes where each agent gets no incentive to unilaterally deviate from her current strategy. While Nash's famous theorem [43] guarantees that any game admits a Nash equilibrium for some kind of randomized (*mixed*) strategies, the computational complexity of computing such an equilibrium was unsettled until very recently. A fundamental result towards answering this question has been established in [25], which showed that, for any normal form game, it is possible to construct a graphical game (where each player has two available strategies and depends on three other players at most) such that we can recover a Nash equilibrium of the original game from any Nash equilibrium of the graphical game; in addition, in [25] it is also observed that the resulting graphical game can eventually be reduced to a normal form game over four players, by "preserving" again Nash equilibria. In fact, by exploiting the ideas in [25], Daskalakis, Goldberg, and Papadimitriou [13] showed that the problem of computing a Nash equilibrium is hard for the class PPAD [48], in the case of games with (at least) four players, and complete for this class in an $\epsilon$-approximated version. The work was then improved to the 3-player case by Chen and Deng [7] and by Daskalakis and Papadimitriou [14] independently. And, finally, Chen and Deng [8] proved that hardness holds even for the case of two-player games. Other questions concerning Nash equilibria (of graphical games) have also recently been faced.

For instance, [53,54] studied relevant problems such as checking whether a given strategy profile is a Nash equilibrium and computing a Nash equilibrium with and without approximation, while [18,1,26] considered the existence of *pure* Nash equilibria, i.e., equilibria where each player must play in a non-aleatory manner.

In this paper, Nash equilibria in graphical games are instead analyzed from a different perspective, by observing that, in many real-world applications, the existence of "any" equilibrium, and the possibility of (approximately) computing it might be not enough. Rather, it is often useful to single out equilibria satisfying some additional, application-oriented constraints. This is, for instance, the case when one looks for equilibria where some players are guaranteed to get at least a certain payoff, or the case when one looks for an equilibrium guaranteeing the maximum payoff to a given player over all the possible strategies (*player optimum*), or the maximum total payoff summed over all the players (*social optimum*), or the maximum payoff for the player getting the minimum one over all the players (*welfare optimum*).

Nash equilibria satisfying additional requirements, called *constrained Nash equilibria* in the following, turned out to be useful in several applicative domains. An example scenario, which received considerable attention in the literature, is given by the selfish routing problem: Each of several agents wants to send a particular amount of traffic along a path from a *source* to a *destination*, thereby defining a game where strategies correspond to paths from the source to the destination, and payoffs are given by (the opposite of) packets delays, as determined by the traffic on network links. In fact, Koutsoupias and Papadimitriou [37] studied this problem in the special case where the network consists of only two nodes and a set of parallel links connecting them; in particular, in order to assess the cost for the lack of a centralized regulating authority, they firstly suggested to investigate the worst-case coordination ratio (called the price of anarchy in [49]), which is the ratio between the *worst possible equilibrium* and the best coordinated routing, i.e., more formally, between the maximum expected latency of traffic through any link (over all Nash equilibria) and the least possible maximum latency that can be guaranteed with some global regulation—see, also, [40,36,12,20,21,19].

Of course, most of the well-known results do not apply to a setting where Nash equilibria have to satisfy additional requirements. Indeed, a constrained Nash equilibrium is even not guaranteed to exist at all, whenever constraints are issued over the game. Thus, it is relevant to know what happens if constraints are added to the game, and whether computing constrained equilibria (if any) is any harder, and under which restrictions it is feasible in polynomial time. For instance, algorithmic issues related to the computation of Nash equilibria for the selfish routing games discussed above have been investigated in [20], where complexity results have been established for several problems arising in this setting, such as the NP-hardness of computing either the best or the worst pure Nash equilibrium, and the #*P*-completeness of computing the social cost of a given mixed Nash equilibrium. In this paper, we aim at conducting this kind of complexity analysis for graphical games.

### *1.2. Earlier results on constrained Nash equilibria*

First results on the computational complexity of equilibria that satisfy some additional requirements have been presented by Gilboa and Zemel [24] in the context of two-player games in normal form. Subsequently, Conitzer and Sandholm [10,11] reconsidered this setting and proposed a single reduction (from satisfiability of Boolean formulas) that sharpened most of the results of [24] and provided novel ones. In fact, it is shown that it is NP-hard to decide the existence of more than one Nash equilibrium, and to determine the existence of a Nash equilibrium where a given player plays (or, does not play) some given strategy. Also, inapproximability results emerged, in particular, for the problems of computing the maximum social welfare and the maximum utility achieved by some player in a Nash equilibrium. And, eventually, it is shown that counting the number of Nash equilibria is #*P*-hard. However, characterizing the complexity of constrained equilibria in graphical games has been left as an open research problem by Conitzer and Sandholm [10,11].

This problem has recently been attacked by Schoenebeck and Vadhan [53,54], who undertake a systematic study of the complexity of Nash equilibria in concisely represented games, by focusing, in particular, on *circuit games* (where payoffs are computed via boolean circuits) and on graphical games. Within these settings, the authors studied the problems of checking whether a given combination of strategies is a Nash equilibrium, of determining the existence of pure Nash equilibria, of computing a Nash equilibrium, and of determining whether a Nash equilibrium exists, achieving certain payoffs guarantees for each player. Interestingly, these results are studied with and without approximation. Among these numerous contributions, an important result (which can be viewed as the counterpart of those by Conitzer and Sandholm [10,11] for two-player games) states that it is NP-complete, for all levels of approximation, to decide the existence of a Nash equilibrium where each player achieves a certain payoff; in particular, hardness is proven for the case where each player has two available strategies, has three neighbors at most, and is required to get, at the equilibrium, its maximum available payoff [53,54].

Classes of games for which constrained Nash equilibria can efficiently be computed (or, approximated) have instead been identified in [34,15,16], by focusing on scenarios where each player has two available strategies. In fact, in their work about the computation of Nash equilibria in graphical games, Kearns, Littman, and Singh [34] considered games with tree-like player interactions (as they appear from the undirected version of the neighborhood relationship), and described a polynomial-time algorithm for computing *approximate* Nash equilibria that can be adapted to work even in the presence of some simple kinds of constraints. In particular, they argued that it is possible to identify approximate Nash equilibria that maximize (i) the payoff of a given player (*player optimum*), (ii) the sum of the payoffs over each player (*social optimum*), and (iii) the smallest payoff over players (*welfare optimum*).

The problem of exactly computing Nash equilibria on trees was, instead, firstly considered in [33], where a data structure called *best response policy* was introduced, which is meant as a way to represent all Nash equilibria of a graphical game. Then, in [15] it is observed that the best response policy has polynomial size as long as the underlying graph is a path, and it is proved that (exact) Nash equilibria can be computed in polynomial time for this class of games. The result in [15] has recently been improved in [16], where it is observed that Nash equilibria guaranteeing certain payoffs to all participants can be computed in polynomial-time on bounded-degree acyclic graphs, where, in addition, the best response policy must have polynomial size. On the other hand, [16] noticed that over 3-player games whose neighborhood relationship is not acyclic, computing Nash equilibria satisfying certain constraints is algebraically infeasible, for it requires strategies over irrational numbers.

### *1.3. Summary of the results*

Even if constrained Nash equilibria are receiving more and more attention in the literature, a complete picture of the complexity issues arising in this setting is missing. In particular, it was not clear how players' interactions are affected by the constraints issued over the game; and, in fact, little was known about constraints involving (arbitrary) sets of players, and equilibria maximizing functions more general than, e.g., the social or the player optimum.

The scope of this paper is precisely to face these research questions in the context of graphical games. To this end, a simple, yet comprehensive, framework for specifying additional properties on Nash equilibria is firstly presented, in which requirements can be defined either as hard constraints or as desiderata (expressed via objective functions to be optimized) on the players' outcomes. Each constraint/desideratum can be defined over an arbitrary number of players (i.e., from one up to all the players); and, in fact, in order to possibly combine the outcome of the various players (of interest in the definition of the requirement) into a single parameter subject to constraints or optimized, *evaluation functions* can be used, i.e., polynomially-computable functions mapping players' payoffs (at a given equilibrium) to rational numbers. A typical example is the evaluation function that computes the sum of players' payoffs at the equilibrium: the resulting value may be optimized (e.g., maximized over all equilibria), or subject to some constraint (e.g., forced to be greater than some given threshold).

According to the kinds of evaluation functions used to specify additional properties at the equilibrium, the following (increasingly stringent) classes of graphical games are defined, which will be more formally discussed in Section 2:

**Arbitrarily Constrained Games [ACGs]**, having no limitation on evaluation functions;
**Polynomially Constrained Games [PCGs]**, where evaluation functions are polynomials in the players' payoffs;
**Linearly Constrained Games [LCGs]**, where evaluation functions linearly depend on the players' payoffs;
**Weakly Constrained Games [WCGs]**, where, in addition to linearity, each constraint is local, for it refers only to a player and her neighborhood.

For each of the above classes, the complexity of two relevant problems has thoroughly been investigated, which are the problem of

(1) Deciding the existence of Nash equilibria satisfying a set of hard constraints; and, the problem of
(2) Computing the Nash equilibrium optimizing an objective function.

All the complexity results refer to the setting where players, their neighborhood, and their utility functions are part of the problem input, while hardness results are established even by assuming that the number of actions for each player is bounded by a fixed constant (namely, two actions available to each player). This is a usual setting in the literature for graphical games. Its dual counterpart is what is assumed for games in normal form, where the set of actions for each player is given in input, but players are fixed. The results of our analysis comprise both bad and good news: on the one hand, new hardness results are proven that shed light on the sources of complexities for these problems; on the other hand, some interesting tractable classes of games are singled out. Results are established for both the case of games with arbitrary neighborhood relationships and the case of games exhibiting restricted kinds of interactions among players.

**Mixed strategies.** In the first part of the paper, we focus on the case of randomized strategies and we depict a precise picture of the complexity of deciding the existence of constrained Nash equilibria and of computing a Nash equilibrium optimizing some given objective function.

A summary of some of our results is illustrated in Fig. 1, which reports, for each class of games, the number of neighbors over each player – which is an indication of the degree of intricacy of players' interactions – sufficing to lead to intractability (NP-hardness). Note that, in the light of the PPAD-completeness in absence of constraints [48], our results evidence that there is a computational price to be paid for checking whether constraints can be satisfied at some equilibrium and for computing a kind of optimum equilibrium. In fact, the figure is meant to synthetically illustrate how hardness results emerge as a combination of players' interactions with the intrinsic complexity of evaluation functions: The more complex evaluation functions are allowed to define constraints, the fewer neighbors are required to gain intractability; for instance, in the case of arbitrarily constrained games, deciding the existence of a constrained Nash equilibrium remains NP-hard even in absence of interactions among the players.

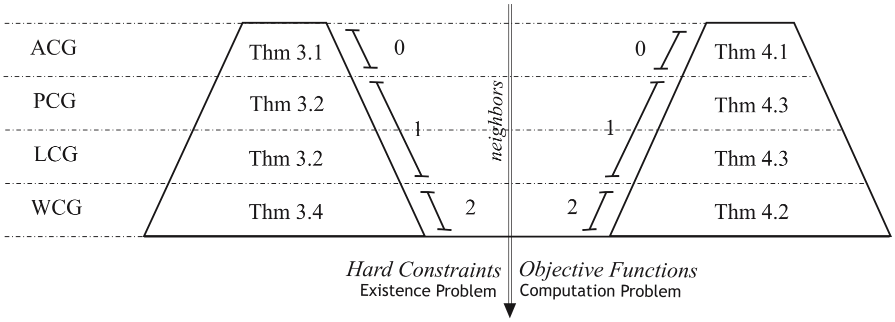



**Fig. 1.** Summary of NP-hardness results for mixed strategies.

Note that results in Fig. 1 are orthogonal w.r.t. those provided by Conitzer and Sandholm [10,11] (in the case of two-player games), for we focused on games where each player has two available strategies only. In addition, note also that Theorem 3.4 partially overlaps with the result of Schoenebeck and Vadhan discussed in Section 1.2: On the one hand, in [53,54] it is shown that NP-hardness holds for all level of approximations. On the other hand, we prove that NP-hardness hold, in absence of approximation, even if there is one constraint on a single player only (rather than on all the players), and if each player has two neighbors at most (rather than three neighbors). In fact, our reduction can be viewed as an extension of the one reported by Schoenebeck and Vadhan, which is designed for relaxing the two conditions above.

In addition to the scenarios illustrated in Fig. 1, two further problems have been analyzed for the case of equilibria optimizing objective functions:

- ▷ Among the various objectives that can be built on top of linear evaluation functions, a prominent role is played in the literature by the *social optimum*, i.e., by the maximization of the total payoff summed over all the players. This optimization function has been studied and its precise complexity has been assessed in terms of computation classes (Theorem 4.4).
- ▷ As an interesting extension of the framework, the case where several objective functions have to be optimized at the same time has been considered as well. In particular, we focus on the well-known notion of *Pareto equilibrium* [2], which is based on the Pareto mechanism, introduced for avoiding the defection of the entire group of players besides the one of single individuals (as in the plain Nash equilibrium). In fact, Pareto equilibria are usually considered in the literature as prototypical kinds of constrained equilibria; for instance, complexity results about Pareto equilibria have been provided by Conitzer and Sandholm [10,11] in the case of (symmetric) two-players games. In this paper, we settle instead their complexity in the graphical games framework (Theorems 4.6 and 4.7).

Note that our analysis was mainly aimed at proving the intractability of constrained Nash equilibria problems and identifying the sources of their intractability. Thus, all proofs are hardness proofs. In particular, we do not provide the corresponding membership proofs (and, hence, the completeness results for the various complexity classes), because it is well-known that, in some cases, Nash equilibria may emerge that are irrational. In fact, facing this representation problem by resorting to approximations of constrained Nash equilibria as discussed in [39,17] is beyond the scope of the present work—this issue is illustrated in more detail in the final section of the paper.

On the positive side, however, we note that if one is interested in tractable classes of constrained Nash equilibria, our results may suggest ways for lowering the complexity by acting on the sources of complexity we pointed out. For instance, one may think of reducing the degree in the neighborhood relationship for a given type of constraints, or of using a weak form of constraints.

**Pure strategies.** In order to depict a complete picture of the impact of constraints on the computation of Nash equilibria, the case of pure strategies is also considered. In this context, deciding the existence of pure Nash equilibria is NP-hard even in absence of constraints [26]. Thus, differently from the case of mixed strategies, our analysis was focused here on isolating tractable classes of constrained games, by complementing the results identified in [26].

In particular, as commonly done in the literature on graphical games, we represent the structure of a game $\mathcal{G}$ by its *dependency graph* $G(\mathcal{G}) = (P, E)$, whose vertices in $P$ coincide with the players of $\mathcal{G}$, and where there is an edge in $\{i, j\} \in E$ if $j$ is a neighbor of $i$, i.e., $j \in Neigh(i)$. In fact, we are using here the definition of dependency graph proposed in [33], for which $G(\mathcal{G})$ is simply the undirected version of the graph encoding the neighborhood relationship of $\mathcal{G}$. Indeed, structural notions such as the *treewidth* [50], leading to tractable classes of games, can more naturally be defined in terms of undirected graphs. Notice, instead, that in the first part of the paper (where the focus is on providing intractability results) the neighborhood relationship is considered, which coincides with the concept of directed dependency graph discussed by various authors (e.g., [53,54,25,13]).

Then, we look for tractable classes by focusing on games with nearly acyclic dependency graphs. In fact, one may compute pure Nash equilibria satisfying the player, social, and welfare optimum, by adapting – with no substantial effort – the algorithm in [34] (see Section 1.2) that was originally conceived for dealing with approximate equilibria in the mixed setting.

In the paper, we improve on this result in two ways: (i) by showing that computing a pure Nash equilibrium is not only feasible in polynomial time, but it is also *highly-parallelizable*, for classes of games having tree-like dependency graphs; and (ii) by showing that our parallelizable algorithm works with a richer variety of additional requirements. Formally:

- ▷ We prove that computing a pure Nash equilibrium satisfying all constraints is feasible in polynomial time for WCGs having tree-like dependency graphs, if there is a bounded number of *smooth* global constraints—very roughly, with polynomially bounded output values. For such games, efficient parallelizable algorithms have been provided (Theorem 5.4).
- ▷ And, we show that even the computation of a pure Nash equilibrium that optimizes a given objective function is feasible in polynomial time, for games having tree-like dependency graphs and a bounded number of smooth constraints (Theorem 5.5).

### 1.4. Organization

The rest of the paper is organized as follows. In Section 2, the framework for dealing with games having different kinds of constraint is illustrated. Based on it, the computational complexity of the problems of deciding the existence of Nash equilibria satisfying additional requirements and of computing a Nash equilibrium optimizing an objective function are studied in Sections 3 and 4, respectively. In Section 5, the setting of pure strategies is considered and new classes of games are identified, whose equilibria can efficiently be computed. Finally, Section 6 draws our conclusions.

## 2. Formal framework

In this section, some basic notions of game theory are first introduced which will be referred to in the paper. Specifically, the concept of equilibrium is defined and illustrated through examples. Then, the basic framework to formalize constraints issued over Nash equilibria is discussed.

### 2.1. Games and Nash equilibria

A *graphical game* $\mathcal{G}$ is a tuple $\langle P, Neigh, Act, U\rangle$, where $P$ is a non-empty set of distinct players, *Neigh* and *Act* are functions, and $U$ is a set of functions. In particular, for each player $i \in P$, the function *Neigh* provides a set of players $Neigh(i) \subseteq P - \{i\}$, called the neighbors of $i$, while $Act(i)$ defines the set of her possible actions, and $U$ contains her utility function $u_i : Act(i) \times_{j\in Neigh(i)} Act(j) \to \mathbb{Q}$, where $\mathbb{Q}$ denotes the set of the rational numbers. Intuitively, $Neigh(i)$ contains the players who potentially matter w.r.t. to $i$'s utility function. Indeed, in general, a player is not directly interested in all other players, and thus her utility function is defined only in terms of the actions played by her neighbors and by herself.

Let $\mathcal{G} = \langle P, Neigh, Act, U\rangle$ be a game. Each player $i$ may choose an action $a \in Act(i)$ with a given probability $p_a$, where $0 \leq p_a \leq 1$. An individual *strategy* for $i$ is any set $S$ such that: for each $a \in Act(i)$, $S$ exactly contains one pair $(p_a, a)$, and $\sum_{(p_a,a)\in S} p_a = 1$. An individual strategy $S$ for a player $i$ is *pure* if it contains one pair $(1, a)$ for some action $a$; in this case, it is convenient to briefly say that *i plays a*. Otherwise, in the general case, the strategy is said *mixed*. The set of all the (individual) strategies for $i$ is denoted by $St(i)$.

For a non-empty set of players $P' \subseteq P$, a *combined strategy* (also, *profile*) for $P'$ is a set containing exactly one strategy for each player in $P'$. Then, $St(P')$ denotes the set of all the combined strategies for the players in $P'$. The combined strategy $\mathbf{x}$ is called *global* if $P' = P$. A global strategy $\mathbf{x}$ is called pure (resp., mixed) if each player in it plays a pure (resp., mixed) strategy.

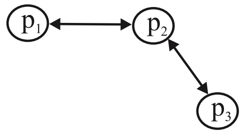
Let $i$ be a player, let $u_i$ be the utility function of $i$, and let $\mathbf{x}$ be a combined strategy for a set of players including $Neigh(i)\cup\{i\}$. Given an action $a$ in $Act(i)$, let us denote by $i_a(\mathbf{x})$ (or even simply by $i_a$, if $\mathbf{x}$ is clear from the context) the probability that player $i$ plays $a$ in the strategy $\mathbf{x}$. With a slight abuse of notation, let us simply denote by $u_i(\mathbf{x})$ the evaluation of $u_i$ on the restriction of $\mathbf{x}$ to its domain, that is, on the restriction to the actions played by $i$ and by her neighbors in $Neigh(i)$. The *payoff* of player $i$ w.r.t. $\mathbf{x}$, denoted by $pay_i(\mathbf{x})$, is the expected value of her utility function given the probability distribution of the actions played by her neighbors in $Neigh(i)$ and by herself, provided their individual strategies in $\mathbf{x}$, i.e., $pay_i(\mathbf{x}) = \mathbf{E}_{\mathbf{x}}[u_i]$. Note that, if $\mathbf{x}$ is a pure combined strategy, then the payoff $pay_i(\mathbf{x})$ of a player $i$ w.r.t. $\mathbf{x}$ simply coincides with $u_i(\mathbf{x})$.

**Example 2.1.** Consider the game $\mathcal{G}_1$ for the players $p_1, p_2$ and $p_3$, with $Neigh(p_1) = \{p_2\}$, $Neigh(p_2) = \{p_1, p_3\}$, and $Neigh(p_3) = \{p_2\}$. Assume that $Act(p_1) = \{a, b\}$, $Act(p_2) = \{c, d\}$, $Act(p_3) = \{e, f\}$, and that the utility functions for the players are those shown in the tables reported in Fig. 2. For instance, given such utility functions, one can easily see that all players get payoff 1 in the global strategy where $p_1$ plays $a$, $p_2$ plays $c$, and $p_3$ plays $e$.

As a further example, consider now a two-player game $\mathcal{G}_2$, whose utility functions are shown in Fig. 3. Here, the game is such that $Act(p_1) = \{t, f\}$ and $Act(p_2) = \{t, f\}$. Let $\mathbf{x}$ be the global strategy such that $p_1$ plays $t$ with probability $\frac{1}{3}$ and $f$ with probability $\frac{2}{3}$, while $p_2$ plays $t$ with probability $\frac{1}{4}$ and $f$ with probability $\frac{3}{4}$. Then,

$$pay_{p_1}(\mathbf{x}) = \tfrac{1}{4}(\tfrac{1}{3} \times 1 + \tfrac{2}{3} \times -1) + \tfrac{3}{4}(\tfrac{1}{3} \times -1 + \tfrac{2}{3} \times 1) = \tfrac{1}{6}$$
$$pay_{p_2}(\mathbf{x}) = \tfrac{1}{3}(\tfrac{1}{4} \times -1 + \tfrac{3}{4} \times 1) + \tfrac{2}{3}(\tfrac{1}{4} \times 1 + \tfrac{3}{4} \times -1) = -\tfrac{1}{6}. \quad \triangleleft$$

| $p_3 \backslash p_2$ | $c$ | $d$ |
|---|---|---|
| $e$ | 1 | 0 |
| $f$ | 0 | 1 |




| $p_1 \backslash p_2$ | $c$ | $d$ |
|---|---|---|
| $a$ | 1 | 2 |
| $b$ | 0 | 1 |

| $p_2 \backslash p_1 p_3$ | $a,e$ | $a,f$ | $b,e$ | $b,f$ |
|---|---|---|---|---|
| $c$ | 1 | 0 | 3 | 2 |
| $d$ | 2 | 1 | 0 | 1 |

**Fig. 2.** Game $\mathcal{G}_1$: utility functions and neighborhood relationship.

| $p_1 \backslash p_2$ | $t$ | $f$ |
|---|---|---|
| $t$ | 1 | −1 |
| $f$ | −1 | 1 |

| $p_2 \backslash p_1$ | $t$ | $f$ |
|---|---|---|
| $t$ | −1 | 1 |
| $f$ | 1 | −1 |

**Fig. 3.** Utility functions for game $\mathcal{G}_2$.

We now formally define the main concept of equilibrium which will be studied in this paper. Let $\mathbf{x}$ be a global strategy, $i$ a player, and $y$ an individual strategy for $i$. Then, denote by $\mathbf{x}_{-i}[y]$ the global strategy where the individual strategy of player $i$ in $\mathbf{x}$ is replaced by $y$.

**Definition 2.2** (*Nash Equilibria*). Let $\mathcal{G} = \langle P, Neigh, Act, U\rangle$ be a game and $\mathbf{x}$ be a global strategy for $\mathcal{G}$. Then, $\mathbf{x}$ is a Nash equilibrium for $\mathcal{G}$ if, $\forall i \in P$, $\nexists y \in St(i)$ such that $pay_i(\mathbf{x}) < pay_i(\mathbf{x}_{-i}[y])$. □

**Example 2.3.** Consider again the two games presented in Example 2.1. As for game $\mathcal{G}_1$, the pure (i.e., non-aleatory) strategy in which $p_1$, $p_2$ and $p_3$ play $a$, $c$ and $e$, respectively, is not a Nash Equilibrium, because $p_2$ finds convenient to deviate from the strategy, by playing $d$. However, the strategy in which $p_1$, $p_2$ and $p_3$ deterministically play $a$, $d$, and $f$, respectively, is a Nash equilibrium. Thus, $\mathcal{G}_1$ admits pure Nash equilibria.

Conversely, $\mathcal{G}_2$ has no pure Nash Equilibrium. However, the strategy in which each player plays $t$ and $f$ with probability $\frac{1}{2}$ is a mixed Nash equilibrium. Indeed, both players get payoff 0 and no one may improve this value by changing her strategy. ◁

### *2.2. Constrained Nash equilibria*

Let $\mathcal{G} = \langle P, Neigh, Act, U\rangle$ be a game and $P'$ be a non-empty subset of the players. An *evaluation function* $f_{P'}$ for players in $P'$ is any polynomial-time computable function that, given any combined strategy $\mathbf{x}$ for the players in $P' \cup \bigcup_{i\in P'} Neigh(i)$, maps the set $\{pay_i(\mathbf{x}) \mid i \in P'\}$ to a rational number. If $P'$ is the whole set of players $P$, we write simply $f$, instead of $f_P$. Note that the domain of an evaluation function $f_{P'}$ is the set of all the possible payoffs associated with players in $P' \cup \bigcup_{i\in P'} Neigh(i)$, for each of their combined strategies; in particular, with respect to computing the value $f_{P'}(\mathbf{x})$, it is completely irrelevant which global strategies may lead to $\mathbf{x}$.

By using evaluation functions, one may define additional properties for Nash equilibria. In particular, we shall consider two basic kinds of requirement:

(1) A *constraint on the payoffs* of the players in $P'$ is an expression $c$ of the form $[f_{P'}\ \mathrm{op}\ k]$, where $k$ is a rational number and $\mathrm{op} \in \{<, >, =, \neq, \leq, \geq\}$. The semantics is as follows: a Nash equilibrium $\mathbf{x}$ satisfies $c$, denoted by $\mathbf{x} \models c$, if $f_{P'}(\mathbf{x})\ \mathrm{op}\ k$. For instance, if op is $<$, then it is required that the evaluation of $f_{P'}$ on the Nash equilibrium $\mathbf{x}$ is less than $k$.
(2) An *objective function constraint* is an expression $o$ of the form $[\mathrm{op}\ f]$, where $\mathrm{op} \in \{\min, \max\}$ and $f$ is an evaluation function. A Nash equilibrium $\mathbf{x}$ is said *optimal* w.r.t. an objective function $o$ of the form $[\min f]$ (resp., $[\max f]$), denoted by $\mathbf{x} \models o$, if there exists no Nash equilibrium $\mathbf{y}$ such that $f(\mathbf{y}) < f(\mathbf{x})$ (resp., $f(\mathbf{y}) > f(\mathbf{x})$).

For any graphical game $\mathcal{G}$, let us denote by $constr(\mathcal{G})$ (resp. $obj(\mathcal{G})$) the set of constraints on the payoffs (resp. the objective constraint) associated with $\mathcal{G}$. A *constrained* Nash equilibrium $\mathbf{x}$ for $\mathcal{G}$ is a Nash equilibrium satisfying each constraint in $constr(\mathcal{G})$ and that is optimal w.r.t. $obj(\mathcal{G})$.

In the rest of the paper, we shall focus on studying the intrinsic complexity of constrained Nash equilibria. In particular, towards a fine-grained analysis, we propose a classification for games based on properties of the evaluation functions on top of which the constraints are built. Formally, an evaluation function $f_{P'}$ is said *polynomial* if it can be written as a polynomial in the payoffs of the various players. In particular, $f_{P'}$ is said *linear* if it is of the form $\Sigma_{i\in P'} w_i \times pay_i(\mathbf{x})$, where $w_i$ is a rational number, for each player $i \in P'$. Moreover, $f_{P'}$ is said *local* if $P' = \{i\}$ for some player $i$. Note that such a constraint can be evaluated by looking at the payoff of $i$ and her neighbors only. Then, we shall investigate the following classes of graphical games:

**Arbitrarily Constrained Games [ACGs]:** This is the most general class of games, where arbitrary evaluation functions are used.
**Polynomially Constrained Games [PCGs]:** In this class, constraints are defined over polynomial evaluation functions only.
**Linearly Constrained Games [LCGs]:** Games in this class are such that constraints are defined over linear evaluation functions only.
**Weakly Constrained (Graphical) Games [WCGs]:** In addition to linearity, each constraint is local.

## 3. Hardness results for constraints on the payoffs

In this section, we start our investigation on the complexity issues related to Nash equilibria by considering the various classes of games defined in Section 2. In particular, we shed light on the intrinsic complexity of deciding the existence of constrained equilibria, by evidencing how the hardness of this task is affected by the kinds of constraints issued over the game and by players' interactions. In particular, in this section we focus on constrained games without any objective function to be optimized.

### 3.1. Arbitrarily constrained games

Our first result is to show that deciding the existence of Nash equilibria in arbitrarily constrained games is NP-hard. The proof evidences that one single arbitrary evaluation function may lead to intractability, even in trivial scenarios where each player has no neighbors at all (so that any global strategy is actually a Nash equilibrium there). Clearly enough, in these cases, strategic interactions do not emerge from the neighborhood relationship, but lay hidden in the constraint at hand.

**Theorem 3.1.** *Deciding whether an arbitrarily constrained graphical game (ACG) $\mathcal{G}$ has constrained Nash equilibria is* NP-*hard, even if:* (i) *there is only one constraint on the payoffs,* (ii) *each player is allowed to play two actions at most, and* (iii) *each player has no neighbors.*

**Proof.** Recall that deciding whether a Boolean formula $\Phi$ over variables $X_1, \ldots, X_n$ is satisfiable is an NP-hard problem [23]. Based on $\Phi$, we build in polynomial time a game $\mathcal{G}(\Phi)$ over the players $x_1, \ldots, x_n$ such that $Neigh(x_i) = \emptyset$, for each $1 \le i \le n$. In fact, each player $x_i$ may choose a strategy in $\{T, F\}$ and, independently on her selection, she gets payoff 1.

Let $\mathbf{x}$ be a global strategy and consider the assignment $\sigma(\mathbf{x})$ for the variables $X_1, \ldots, X_n$, where $X_i$ evaluates true (resp., false) in $\sigma(\mathbf{x})$ if the corresponding player $x_i$ plays $T$ in $\mathbf{x}$ with probability $x_{i_T} > \frac{1}{2}$ (resp., $x_{i_T} \le \frac{1}{2}$). Moreover, consider the constraint $c : [f^c = 1]$, where:

$$f^c(\mathbf{x}) = \begin{cases} 1 & \text{if } \sigma(\mathbf{x}) \text{ satisfies } \Phi \text{, and} \\ 0 & \text{otherwise.} \end{cases}$$

Then, it is immediate to check there is a constrained equilibrium for $\mathcal{G}(\Phi)$ if and only if $\Phi$ is satisfiable. Note that players $x_1, \ldots, x_n$ do not interact at all in this game, but through the constraint $c$. $\Box$

### 3.2. Polynomially and linearly constrained games

We now turn to a more interesting class of games, where constraints are not completely arbitrary. Here, intractability results are not intrinsic in the kinds of imposed constraint. Rather, the source of intractability is rooted in the interplay between constraints and global strategies at equilibria.

In fact, we shall show that deciding the existence of Nash equilibria in linearly constrained games is NP-hard, as long as there is some "minimal" form of interaction among the agents, as formalized below.

**Theorem 3.2.** *Deciding whether a linearly (and, hence, polynomially) constrained graphical game (LCG) $\mathcal{G}$ has constrained Nash equilibria is* NP-*hard, even if:* (i) *there are only three constraints on the payoffs,* (ii) *each player is allowed to play two actions at most,* (*iii*) *each player has one neighbor at most, and* (*iv*) *the neighborhood relationship is acyclic.*

**Proof.** Recall that an instance *ec* of the problem EXACT COVER (BY 3-SETS) is given by a set of elements $I_1, \ldots, I_{3n}$, and by some sets $S_1, \ldots, S_m$, each one containing exactly three elements in $\{I_1, \ldots, I_{3n}\}$. Deciding whether there is an exact cover of these elements, i.e., a set $\mathcal{C} \subseteq \{S_1, \ldots, S_m\}$ such that $\bigcup_{S_i \in \mathcal{C}} S_i = \{I_1, \ldots, I_{3n}\}$ and $S_i \cap S_j = \emptyset$, for each $S_i, S_j \in \mathcal{C}$ with $i \ne j$, is NP-complete [23]. Note that, for any exact cover $\mathcal{C}$, it holds that $|\mathcal{C}| = n$.

Consider the game $\mathcal{G}_{ec}$ defined as follows. The set of players $P_{ec}$ exactly contains the three players $r^k$, $s^k$ and $p^k$ for each set $S_k$, and the player $p^k_\alpha$ for each item $I_\alpha$ in $S_k$. Let $I_i$, $I_j$ and $I_h$ be the three elements in $S_k$ with $i < j < h$, and define the neighborhood as follows: $Neigh(r^k) = \emptyset$, $Neigh(s^k) = \{r^k\}$ $Neigh(p^k) = \{s^k\}$, $Neigh(p^k_i) = \{p^k\}$, $Neigh(p^k_j) = \{p^k_i\}$, and $Neigh(p^k_h)$

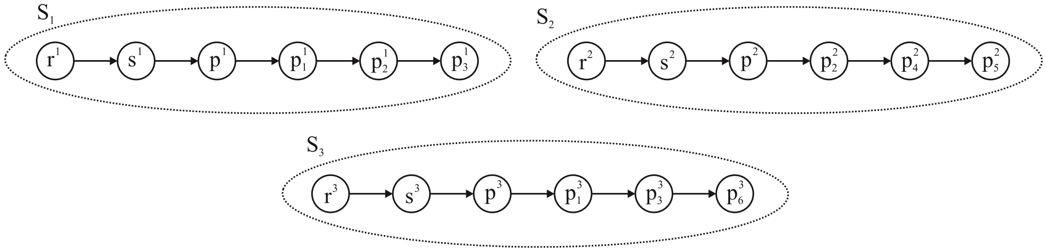



**Fig. 4.** Reduction from EXACT COVER.

$= \{p_j^k\}$. As an example construction, the neighborhood relationship of the game associated with the sets $S_1 = \{I_1, I_2, I_3\}$, $S_2 = \{I_2, I_4, I_5\}$, and $S_3 = \{I_1, I_3, I_6\}$ is illustrated in Fig. 4.

Each player in $\mathcal{G}_{ec}$ may choose a strategy from the set $\{IN, OUT\}$. In particular, let $\bar{\mathbf{x}}$ be a global pure strategy, and let $r^k$, $s^k$, $p^k$, $p_i^k$, $p_j^k$, and $p_h^k$ be the players associated with the set $S_k$. Then, the utility function $u_{r^k}$ is such that $pay_{r^k}(\bar{\mathbf{x}}) = 1$, no matter of $\bar{\mathbf{x}}$. Moreover, $u_p$ with $p \in \{s^k, p^k, p_i^k, p_j^k, p_h^k\}$ is such that:

- $pay_p(\bar{\mathbf{x}}) = 1$, if $p$ plays *IN* and her neighbor plays *IN*;
- $pay_p(\bar{\mathbf{x}}) = 0$, in all other cases.

Recall that for any player $i$ and action $a$, $i_a(\mathbf{x})$ denotes the probability that $i$ plays $a$ in a given strategy $\mathbf{x}$. Then, towards establishing the result, we claim that, whenever $r_{IN}^k(\mathbf{x}) > 0$ in a given (possibly mixed) Nash equilibrium $\mathbf{x}$, it must also be the case that $p_{IN}(\mathbf{x}) = 1$, for each player $p \in \{s^k, p^k, p_i^k, p_j^k, p_h^k\}$. Indeed, for each such player $p$ and for the (unique) player $q \in Neigh(p)$, it holds that: $pay_p(\mathbf{x}) = p_{IN}(\mathbf{x}) \times q_{IN}(\mathbf{x})$. Hence, whenever $q_{IN}(\mathbf{x}) > 0$, the maximum payoff of $p$ is achieved for $p_{IN}(\mathbf{x}) = 1$. Thus, the claim follows by starting with $q = r^k$, for which $s_{IN}^k(\mathbf{x}) = 1$ holds, and by iteratively applying this argument for $q = s^k$, $q = p^k$, $q = p_i^k$, and $q = p_j^k$.

Let $\mu = m + 1$ and let $constr(\mathcal{G}_{ec})$ contains the following constraints:

$$c_1: \quad [f^{c_1} = n], \text{ where } f^{c_1}(\mathbf{x}) = \sum_{k=1}^{m} pay_{s^k}(\mathbf{x})$$

$$c_2: \quad [f^{c_2} = 3 \times n], \text{ where } f^{c_2}(\mathbf{x}) = \sum_{i=1}^{3n} \left( \sum_{p_j^k | j=i} pay_{p_j^k}(\mathbf{x}) \right)$$

$$c_3: \quad \left[ f^{c_3} = \sum_{i=1}^{3n} \mu^i \right], \text{ where } f^{c_3}(\mathbf{x}) = \sum_{i=1}^{3n} \left( \mu^i \times \sum_{p_j^k | j=i} pay_{p_j^k}(\mathbf{x}) \right).$$

Note that $\mathcal{G}_{ec}$ and $constr(\mathcal{G}_{ec})$ can be built in polynomial time from the given instance of EXACT-COVER, and that $f^{c_1}$, $f^{c_2}$, and $f^{c_3}$ are linear evaluation functions that can be evaluated in polynomial time. In particular, all constraints are linear, and encoding any coefficient $\mu^i$ in the third linear combination (with $1 \le i \le 3n$) requires polynomially many bits.

Thus, to conclude the proof, we need to show that: *there exists an exact cover of ec* $\Leftrightarrow$ $\mathcal{G}_{ec}$ *admits a Nash equilibrium satisfying* $constr(\mathcal{G}_{ec})$.

($\Rightarrow$) Assume there exists an exact cover, say $\mathcal{C}$. Consider the global strategy $\mathbf{x}$ for $\mathcal{G}_{ec}$ where each player in $P_{ec}$ chooses her action according to $\mathcal{C}$, i.e., for each set $S_k = \{I_i, I_j, I_h\}$ in $\mathcal{C}$, players $r^k$, $s^k$, $p^k$, $p_i^k$, $p_j^k$ and $p_h^k$ play *IN*, whereas all the players associated with any set $S_{k'} \notin \mathcal{C}$ play *OUT*. Note that each player associated with a set $S_k \in \mathcal{C}$ gets her maximum available payoff and, hence, she gets no incentive to deviate from $\mathbf{x}$. Instead, each player associated with a set $S_{k'} \notin \mathcal{C}$ gets payoff 0 (but for $r^{k'}$, getting payoff 1) independently on her actual strategy, since her neighbor plays *OUT*; thus, she has no chances to get a better payoff. So, $\mathbf{x}$ is a Nash equilibrium.

It remains to show that $c_1$, $c_2$, and $c_3$ are satisfied by $\mathbf{x}$. To this end, note that for each index $i \in \{1, \ldots, 3n\}$, there exists exactly one set $S_{\bar{k}} \in \mathcal{C}$ with $I_i \in S_{\bar{k}}$. Hence, player $p_i^{\bar{k}}$ gets payoff 1, whereas all the other players of the form $p_i^{k'}$, with $k' \neq \bar{k}$, get payoff 0. Thus, $\Sigma_{p_j^k | j=i} pay_{p_j^k}(\mathbf{x}) = 1$, which entails that $c_2$ and $c_3$ are satisfied.

In addition, player $s^k$ gets payoff 1 if $S_k$ is in $\mathcal{C}$; and, she gets payoff 0 if $S_k$ does not occur in $\mathcal{C}$. Since exactly $n$ sets are in $\mathcal{C}$, $c_1$ is satisfied as well.

($\Leftarrow$) Consider a Nash equilibrium $\mathbf{x}$ that satisfies $constr(\mathcal{G}_{ec})$. We first need to state some important properties of $\mathbf{x}$.

**Claim 3.2.1.** *The following conditions hold:* (a) $|\{k \mid p_{IN}^k(\mathbf{x}) = 1\}| = |\{k \mid p_{IN}^k(\mathbf{x}) > 0\}| = n$*; and,* (b) *the restriction of* $\mathbf{x}$ *to the players in* $\bigcup_{k=1}^{m} \{p^k, p_i^k, p_j^k, p_h^k\}$ *is a combined pure strategy.*

We start by proving (a). To this end, observe that in order to satisfy $c_1$, it must be the case that $n = \sum_{k=1}^{m} pay_{s^k}(\mathbf{x})$. Then, note that, by construction,

$$n = \sum_{k=1}^{m} pay_{s^k}(\mathbf{x}) = \sum_{k=1}^{m} s_{IN}^k(\mathbf{x}) \times r_{IN}^k(\mathbf{x}) = \sum_{k=1}^{m} r_{IN}^k(\mathbf{x}),$$

where the latter equality holds since $r_{IN}^k(\mathbf{x}) \neq \emptyset$ implies $s_{IN}^k(\mathbf{x}) = 1$ at the equilibrium $\mathbf{x}$. Thus, in order to have $n = \sum_{k=1}^{m} r_{IN}^k(\mathbf{x})$, it must be the case that at least $n$ players of the form $r^k$ do not play *OUT*, i.e., that $|\{k \mid r_{IN}^k(\mathbf{x}) > 0\}| \geq n$. Eventually, this entails that at least $n$ players of the form $s^k$, and then of the form $p^k$, play *IN* with probability 1 at the equilibrium $\mathbf{x}$. Thus, $|\{k \mid p_{IN}^k(\mathbf{x}) = 1\}| \geq n$.

Recall now that $p_{IN}^k(\mathbf{x}) = 1$ implies that $q_{IN}(\mathbf{x}) = 1$ holds at the equilibrium (and, hence, $pay_{q^k}(\mathbf{x}) = 1$), for each $q \in \{p_i^k, p_j^k, p_h^k\}$. Let $\bar{\mathcal{Q}}$ be the set of players $\{p_\ell^k \mid p_{IN}^k(\mathbf{x}) < 1\}$. Then, we have

$$\sum_{i=1}^{3n} \left( \sum_{p_j^k | j=i} pay_{p_j^k}(\mathbf{x}) \right) = 3 \times |\{k \mid p_{IN}^k(\mathbf{x}) = 1\}| + \sum_{q \in \bar{\mathcal{Q}}} pay_q(\mathbf{x}).$$

In order to satisfy $c_2$, the above relationships (combined with the fact that $|\{k \mid p_{IN}^k(\mathbf{x}) = 1\}| \geq n$ holds) immediately entails that $|\{k \mid p_{IN}^k(\mathbf{x}) = 1\}| = n$, and hence that $\sum_{q \in \bar{\mathcal{Q}}} pay_q(\mathbf{x}) = 0$. In particular, note that no player of the form $p^k$ exists such that $0 < p_{IN}^k(x) < 1$, for otherwise $pay_{p_i^k}(\mathbf{x}) = p_{IN}^k(\mathbf{x}) > 0$ would hold, thereby leading to $\sum_{q \in \bar{\mathcal{Q}}} pay_q(\mathbf{x}) > 0$. Thus, $|\{k \mid p_{IN}^k(\mathbf{x}) = 1\}| = |\{k \mid p_{IN}^k(\mathbf{x}) > 0\}| = n$, which proves (a).

Moreover, note that $\sum_{q \in \bar{\mathcal{Q}}} pay_q(\mathbf{x}) = 0$ entails $pay_q(\mathbf{x}) = 0$, for each $q \in \bar{\mathcal{Q}}$. Combined with (a), this means that for each $k \in \{1, \ldots, m\}$ and for each $q \in \{p^k, p_i^k, p_j^k, p_h^k\}$, either $q_{IN}(x) = 0$ or $q_{IN}(\mathbf{x}) = 1$ holds. In other words, the restriction of $\mathbf{x}$ to the players in $\bigcup_{k=1}^{m} \{p^k, p_i^k, p_j^k, p_h^k\}$ is a combined pure strategy, which proves (b).

**Claim 3.2.2.** *$\Sigma_{p_j^k | j=i} pay_{p_j^k}(\mathbf{x}) = 1$, for each $i \in \{1, \ldots, 3n\}$.*

From Claim 3.2.1.(b), $\Sigma_{p_j^k | j=i} pay_{p_j^k}(\mathbf{x})$ is a natural number in $\{0, \ldots, m\}$, for any $1 \leq i \leq 3n$. Then, the claim follows from the fact that constraint $c_3$ is satisfied at $\mathbf{x}$. Indeed, recall that $\mu = m + 1$ and $\Sigma_{p_j^k | j=i} pay_{p_j^k}(\mathbf{x}) \leq m$ (for each $1 \leq i \leq 3n$). Then, any value of $f^{c_3}(\mathbf{x})$ can be viewed as a base $\mu$ number, whose digits for different powers of $\mu$ do not interact with each other. It follows that the only way to get the prescribed value $\Sigma_{i=1}^{3n} \mu^i$ for $f^{c_3}(\mathbf{x})$ is when all these digits are equal to 1.

More formally, assume for the sake of contradiction that there is an index $\bar{i} \in \{1, \ldots, 3n\}$ such that $\Sigma_{p_j^k | j=\bar{i}} pay_{p_j^k}(\mathbf{x}) \neq 1$ and that $\Sigma_{p_j^k | j=i} pay_{p_j^k}(\mathbf{x}) = 1$, for each $\bar{i} < i \leq 3n$. Then, $c_3$ implies the following equality:

$$\sum_{i=1}^{\bar{i}-1} \left( \mu^i \times \sum_{p_j^k | j=i} pay_{p_j^k}(\mathbf{x}) \right) + \mu^{\bar{i}} \times \sum_{p_j^k | j=\bar{i}} pay_{p_j^k}(\mathbf{x}) = \sum_{i=1}^{\bar{i}-1} \mu^i + \mu^{\bar{i}}. \tag{1}$$

Consider the term $\Sigma_{p_j^k | j=\bar{i}} pay_{p_j^k}(\mathbf{x})$. Clearly the above equality cannot hold if this term is 0, and we know that it is different from 1, by the hypothesis. Then, because of Claim 3.2.1.(b), it must be a natural number in $\{2, \ldots, m\}$. That is, $\Sigma_{p_j^k | j=\bar{i}} pay_{p_j^k}(\mathbf{x}) \geq 2$. By substituting the above inequality in Eq. (1), we get $\Sigma_{i=1}^{\bar{i}-1} \mu^i + \mu^{\bar{i}} \geq 2 \times \mu^{\bar{i}}$, and thus $\mu^{\bar{i}} \leq \Sigma_{i=1}^{\bar{i}-1} \mu^i$, which is impossible, because $\mu > 1$.

Armed with these two properties for the constrained Nash equilibrium $\mathbf{x}$, we can now show how to build an exact cover. Let $\mathcal{C}$ be the set $\{S_k \mid p_{IN}^k(\mathbf{x}) = 1\}$. We claim that $\mathcal{C}$ is an exact cover. To this end, observe first that because of Claim 3.2.1.(a), $|\mathcal{C}| = n$ holds. Thus, it suffices to show that $S_{k'} \cap S_{k''} = \emptyset$ for each $S_{k'}, S_{k''} \in \mathcal{C}$ with $k' \neq k''$. Indeed, assume for the sake of contradiction that an element $I_i$ exists such that $I_i \in S_{k'} \cap S_{k''}$ for $k' \neq k''$. Since by construction of $\mathcal{C}$, $s_{IN}^{k'}(\mathbf{x}) = s_{IN}^{k''}(\mathbf{x}) = 1$, it is the case that both $p_i^{k'}$ and $p_i^{k''}$ in turn play *IN* in $\mathbf{x}$, thereby getting payoff 1. Thus, $\Sigma_{p_j^k | j=i} pay_{p_j^k}(\mathbf{x}) > 1$, which contradicts Claim 3.2.2. □

As a concluding remark, we note that the above intractability result is established by means of a construction over a game whose neighborhood relationship is acyclic. Actually, for such kinds of game, it is possible to compute an arbitrary Nash equilibrium in polynomial-time, by selecting a strategy for all the players who are not influenced by the choices of other players, by propagating this strategy to their neighbors (as to filter strategies that cannot lead to equilibria), and by iterating the process starting with such neighbors. Therefore, in this case, constraints are the main source of intractability.

### 3.3. Weakly constrained games

In this section, we shall study the case of weakly constrained games where, differently from LCGs, evaluation functions are now required not only to be linear but also to be local. These games are thus simpler than the LCGs, because here constraints may only involve players that are neighbors of each other. It follows that there is no way for such constraints to encode any further kind of (directed) relationship among players outside their neighborhood. However, we shall show that dealing with Nash equilibria is still difficult, though the source of complexity is now changed, as the intricacy of the neighborhood relationship plays a more crucial role.

Technically, our result will exploit a construction relating games and Boolean formulas. In this respect, recall from Section 1 that our setting is orthogonal to that of Conitzer and Sandholm [10,11], which considered a different construction designed for two-player games with an unbounded number of available actions. In fact, we have already stated that our reduction can be viewed, instead, as an extension of the one reported by Schoenebeck and Vadhan [53,54], in the light of proving hardness for games where there is one constraint on a single player only, and where each player has two neighbors at most. For the sake of completeness, we observe, here, that another construction fitting the context of weakly constrained games and relating Boolean formulas and games has recently been proposed in [5], for the slightly different problem of deciding whether a graphical game has more than one Nash equilibrium. Thus, in principle, we may think of adapting the reduction in [5] to our ends, i.e., to the problem of deciding whether a constrained Nash equilibrium exists at all. Yet, we prefer to resort to the construction we originally conceived in [31,30] (where preliminary versions of part of the present work appeared), firstly because it is antecedent to the work of [5] and secondly because it allows us to obtain tighter results, given that the reduction in [5] deals with games where players have up to five neighbors.

#### 3.3.1. Boolean formulas and games

Recall that the following 3SAT problem is NP-hard [23]: decide whether a Boolean formula in conjunctive normal form $\Phi = c_1 \wedge \cdots \wedge c_m$ over variables $X_1, \ldots, X_n$ is satisfiable, i.e., decide whether there are truth assignments to the variables making each clause $c_j$ true, where each clause contains three distinct (possibly negated) variables at most.

W.l.o.g, assume that $\Phi$ contains at least one clause and one variable, and that the number of clauses is such that there exists a positive integer $l$ with $m = 2^l$. In fact, for the latter assumption, if $m$ is such that $2^{l-1} < m < 2^l$, then one can construct in polynomial time a new Boolean formula $\Phi'$ by adding $2^l - m$ new clauses, each one containing a fresh variable. Obviously, these clauses are trivially satisfiable, and hence $\Phi$ and $\Phi'$ are equivalent.

Let us define a game $\mathcal{G}(\Phi)$ such that: The players belong to six pairwise disjoint sets $P_v$, $P_{v'}$, $P_{v''}$, $P_c$, $P_{\bar{c}}$, and $P_t$ plus one distinguished player $E$. The set $P_v$ (resp. $P_{v'}$, $P_{v''}$) contains exactly one player, say $x_i$ (resp. $x_i'$, $x_i''$) for each variable $X_i$ in $\Phi$. Players in $P_c$ are in one-to-one correspondence with the clauses. For each clause $c_j$ containing exactly three variables, $P_{\bar{c}}$ contains the player $\bar{c}_j$ and no other player is in $P_{\bar{c}}$. For each player $x_i \in P_v$, her set of neighbors $Neigh(x_i)$ consists of the players $x_i'$ and $x_i''$, for which $Neigh(x_i') = \{x_i''\}$ and $Neigh(x_i'') = \{x_i'\}$ hold. Let $c_j$ be a clause over the three variables $x_{i_1}, x_{i_2}$, and $x_{i_3}$ (with $i_1 < i_2 < i_3$). Then, $Neigh(c_j) = \{x_{i_1}, \bar{c}_j\}$ and $Neigh(\bar{c}_j) = \{x_{i_2}, x_{i_3}\}$—intuitively, $\bar{c}_j$ is a subclause of $c_j$. Moreover, each variable $x_i$, occurring in a clause $c_j$ together with another variable at most, is in the set $Neigh(c_j)$, and no other players are in this set.

Players in $P_t$ and player $E$ are such that the subgraph of the neighborhood relationship of $\mathcal{G}(\Phi)$ induced by the nodes in $P_c \cup P_t \cup E$, is a complete binary tree rooted at $E$, having as leaves the players in $P_c$. For each player $t$ in $P_t \cup \{E\}$, $Neigh(t)$ consists of the set of the two players that are children of $t$ in tree induced over $G(\mathcal{G}(\Phi))$. Notice that, by construction, $E$ is not a neighbor of any other player, and her choices do not affect the payoffs of the other players in the game. Moreover each player has two neighbors at most.

As an example construction, consider the Boolean formula $\bar{\Phi} = \exists X_1 \ldots X_8 (X_1 \vee X_2) \wedge (X_1 \vee X_3) \wedge (\neg X_1 \vee \neg X_3 \vee \neg X_4) \wedge (X_4) \wedge (\neg X_5 \vee \neg X_6) \wedge (\neg X_4 \vee X_6) \wedge (X_6 \vee X_7) \wedge (X_8)$. Fig. 5 gives on the left a graphical representation of the neighborhood relationship in the game $\mathcal{G}(\bar{\Phi})$ associated with $\bar{\Phi}$.

Let $\{T, F\}$ (read *true* and *false*, respectively) be the set of possible actions for each player in $\mathcal{G}(\Phi)$, and let $\bar{\mathbf{x}}$ be a global pure strategy. Utility functions are defined as follows.

For each variable $X_i$ of the formula $\Phi$, we have three players $x_i$, $x_i'$ and $x_i''$, whose utility functions are shown in tabular form in Fig. 6. Intuitively, we would like to establish a correspondence between truth-value assignments of the formula and strategies chosen by players: the choice $T$ (resp. $F$) of player $x_i \in P_v$ means that the corresponding variable $X_i$ is assigned true (resp. false).

For the utility functions of players corresponding to the clauses of the formula, we distinguish two cases depending on the number of clause literals: Let $c$ be a player in $P_c$ whose corresponding clause in $\Phi$ has two literals at most, or a player in $P_{\bar{c}}$ corresponding to a two-literal subclause of some clause of $\Phi$. Then, her utility function $u_c$ is such that:

**(C-i)** $pay_c(\bar{\mathbf{x}}) = 1$, if the players in $Neigh(c)$ make the corresponding clause/subclause true (resp. false), and $c$ plays $T$ (resp. $F$);

**(C-ii)** $pay_c(\bar{\mathbf{x}}) = 0$, in all the other cases.

Moreover, let $c$ be a player in $P_c$ whose corresponding clause in $\Phi$ has exactly three literals. The neighbors of this player are one variable player and one subclause player in $P_{\bar{c}}$. Then, her utility function $u_c$ is such that:

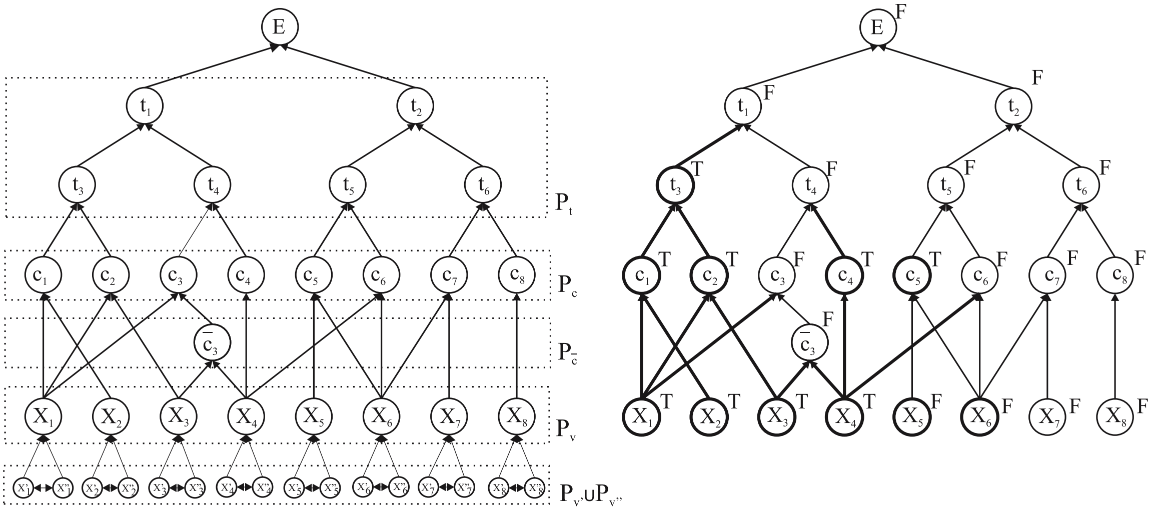



**Fig. 5.** Left: Neighborhood relationship in $\mathcal{G}(\bar{\Phi})$. Right: A Nash equilibrium for $\mathcal{G}(\bar{\Phi})$.

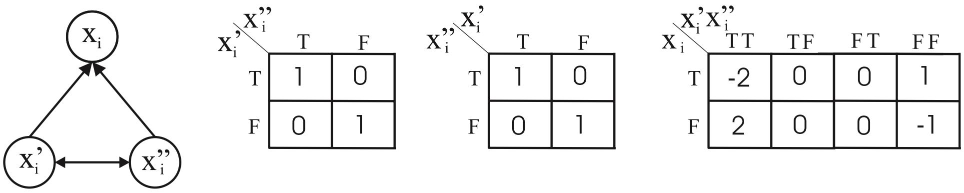



**Fig. 6.** Utility functions for the game $\mathcal{G}(\Phi)$.

**(C-i)** $pay_c(\bar{\mathbf{x}}) = 1$, if the variable player in $Neigh(c)$ does not make the corresponding clause true, $\bar{c} \in Neigh(c)$ plays $F$, and $c$ plays $F$; or, if either the variable player in $Neigh(c)$ makes the clause true or $\bar{c} \in Neigh(c)$ plays $T$, and $c$ plays $T$;
**(C-ii)** $pay_c(\bar{\mathbf{x}}) = 0$, in all the other cases.

Each player $t \in P_t$ acts as an AND-gate on the truth values coming from her neighbors. Her utility function $u_t$ is such that:

**(T-i)** $pay_t(\bar{\mathbf{x}}) = 1$, if either $t$ plays $T$ and all the players in $Neigh(t)$ play $T$, or $t$ plays $F$ and there exists a player in $Neigh(t)$ playing $F$;
**(T-ii)** $pay_t(\bar{\mathbf{x}}) = 0$, in all the other cases.

Finally, player $E$ is responsible for the evaluation of $\Phi$ on the truth assignment induced by the players in $P_v$, and she gets a higher payoff in the case $\Phi$ is satisfied. More precisely, her utility function is such that:

**(E-i)** $pay_E(\bar{\mathbf{x}}) = 2$, if $E$ plays $T$ and all the players in $Neigh(E)$ play $T$;
**(E-ii)** $pay_E(\bar{\mathbf{x}}) = 1$, if $E$ plays $F$ and there is a player in $Neigh(E)$ playing $F$;
**(E-iii)** $pay_E(\bar{\mathbf{x}}) = 0$, in all the other cases.

Let $\bar{\mathbf{x}}$ be a global pure strategy for $\mathcal{G}(\Phi)$ such that each player $x_i \in P_v$ plays either $T$ or $F$ with probability 1 in $\mathbf{x}$. Note that the choices for $\bar{\mathbf{x}}$ uniquely identify a truth-value assignment for $\Phi$, that is denoted by $\sigma^{\bar{\mathbf{x}}}$. The following result states a useful relationship between satisfying truth-value assignments for $\Phi$ and Nash equilibria of $\mathcal{G}(\Phi)$.

**Lemma 3.3.** *Let $\Phi$ be a Boolean formula. Then, $\Phi$ is satisfiable $\Leftrightarrow$ there exists a Nash equilibrium $\mathbf{x}$ for $\mathcal{G}(\Phi)$ such that $E$ plays $T$ with probability 1 in $\mathbf{x}$.*

**Proof.** Beforehand, a number of properties of Nash equilibria of $\mathcal{G}(\Phi)$ are shown, which will be useful for the proof. The first one regards the gadget shown in Fig. 6 and evidences that players $x'_i$ and $x''_i$ are designed in such a way that player $x_i$ will eventually choose a pure strategy in any Nash equilibrium. This is a very important feature of game $\mathcal{G}(\Phi)$, since it allows us to reason mostly about boolean values rather than about probabilities. Formally:

**Property A.** *If a global strategy $\mathbf{x}$ is a Nash equilibrium for $\mathcal{G}(\Phi)$ then, for each player $x_i \in P_v$, $x_i$ plays either $T$ or $F$ with probability 1 in $\mathbf{x}$.*

In order to prove the claim, let us first calculate the expected payoffs of the players $x'_i$ and $x''_i$ in any global strategy $\mathbf{x}$:

- $pay_{x'_i}(\mathbf{x}) = x'_{i_T}x''_{i_T} + x'_{i_F}x''_{i_F} = x'_{i_T}x''_{i_T} + (1 - x'_{i_T})(1 - x''_{i_T})$;
- $pay_{x''_i}(\mathbf{x}) = x'_{i_T}x''_{i_T} + x'_{i_F}x''_{i_F} = x'_{i_T}x''_{i_T} + (1 - x'_{i_T})(1 - x''_{i_T})$

where the identity $x'_{i_T} + x'_{i_F} = 1$ (resp. $x''_{i_T} + x''_{i_F} = 1$) holds, because $T$ and $F$ are the only two available actions for player $x'_i$ (resp. $x''_i$).

Then, it can be easily seen that the only possible strategies for $x_i'$ and $x_i''$ at the equilibrium $\mathbf{x}$ are:

- $s_1$, in which they both play $T$ with probability 0 getting payoff 1;
- $s_2$, in which they both play $T$ with probability 1 getting payoff 1; and
- $s_3$, in which they both play $T$ with probability $\frac{1}{2}$ getting payoff $\frac{1}{2}$.

Similarly, for the expected payoff of player $x_i$, we have

$$\begin{aligned} pay_{x_i}(\mathbf{x}) &= -2x_{i_T}x'_{i_T}x''_{i_T} + x_{i_T}x'_{i_F}x''_{i_F} + 2x_{i_F}x'_{i_T}x''_{i_T} - x_{i_F}x'_{i_F}x''_{i_F} \\ &= (1-2x_{i_T})(2x'_{i_T}x''_{i_T} - (1-x'_{i_T})(1-x''_{i_T})), \end{aligned}$$

where the identity $x_{i_T} + x_{i_F} = 1$ has been used.

By letting $\alpha = 2x'_{i_T}x''_{i_T} - (1-x'_{i_T})(1-x''_{i_T})$, we then get $pay_{x_i}(\mathbf{x}) = (1-2x_{i_T})\alpha$. And, eventually, we can distinguish the following three cases, depending on the strategies of players $x_i'$ and $x_i''$:

(1) Players $x_i'$ and $x_i''$ choose $s_1$ in $\mathbf{x}$: Then $\alpha = -1$, and player $x_i$ finds convenient to play $T$ with probability 1, getting payoff 1;
(2) Players $x_i'$ and $x_i''$ choose $s_2$ in $\mathbf{x}$: Then $\alpha = 2$, and player $x_i$ finds convenient to play $T$ with probability 0, getting payoff 2;
(3) Players $x_i'$ and $x_i''$ choose $s_3$ in $\mathbf{x}$: Then $\alpha = \frac{1}{4}$, and player $x_i$ finds convenient to play $T$ with probability 0, getting payoff $\frac{1}{4}$.

Then, since $\mathbf{x}$ is a Nash equilibrium, we can conclude that each player $x_i \in P_v$ plays either $T$ (case 1 above) or $F$ (case 2 and case 3 above) in $\mathbf{x}$.

Intuitively, Property A above tells that players in $P_v$ encode an assignment for $\Phi$ in any Nash equilibrium $\mathbf{x}$ of $\mathcal{G}(\Phi)$. Actually, this correspondence is one-to-one, as shown below.

**Property B.** *Let $\sigma$ be a truth-value assignment for $\Phi$. Then, there exists a pure Nash equilibrium $\mathbf{x}$ such that $\sigma^{\mathbf{x}} = \sigma$.*

Given the truth assignment $\sigma$, let us consider the global *pure* strategy $\mathbf{x}$ for $\mathcal{G}(\Phi)$ where: each player in $P_v$ chooses its individual strategy according to $\sigma^{\mathbf{x}} = \sigma$; each pair of players of the form $x_i'$ and $x_i''$ choose strategy $s_1$ or $s_2$ (see Property A above) depending on whether $x_i$ plays $T$ or $F$ in $\mathbf{x}$, respectively; each player in $P_c \cup P_{\bar{c}}$ applies the rule **(C-i)**, i.e., she correctly evaluates the clause; and all players in $P_t \cup \{E\}$ act as AND-gates on the inputs of their children, according to the rules **(T-i)**, **(E-i)** and **(E-ii)**. As an example of this construction, Fig. 5 evidences the strategy for $\mathcal{G}(\bar{\Phi})$ associated with the truth assignment $X_1 = X_2 = X_3 = X_4 = T$ and $X_5 = X_6 = X_7 = X_8 = F$.

Then, it is easy to see that $\mathbf{x}$ is a Nash equilibrium since each player gets in $\mathbf{x}$ her maximum available payoff, given the strategies played by all the other players in the game.

Finally, the following property characterizes equilibria associated to truth-value *satisfying* assignments.

**Property C.** *Let $\sigma$ be a truth-value assignment for $\Phi$, and let $\mathbf{x}$ be a pure Nash equilibrium for $\mathcal{G}(\Phi)$ such that $\sigma^{\mathbf{x}} = \sigma$. If $\sigma$ is satisfying, then $E$ plays $T$ with probability 1 in $\mathbf{x}$; otherwise, $E$ plays $T$ with probability 0 in $\mathbf{x}$.*

By Property B above, a Nash equilibrium $\mathbf{x}$ such that $\sigma^{\mathbf{x}} = \sigma$ always exists. Moreover, for any other Nash equilibrium $\mathbf{x}'$ such that $\sigma^{\mathbf{x}'} = \sigma$, all the players in $P_c \cup P_{\bar{c}} \cup P_t \cup \{E\}$ must play the same action as in $\mathbf{x}$. Indeed, if any of the players in $P_c \cup P_{\bar{c}}$ does not apply rule **(C-i)** in $\mathbf{x}'$, she would get payoff 0 in $\mathbf{x}'$ and gets an incentive in deviating from $\mathbf{x}'$ (by applying **(C-i)**), which is impossible. Then, players in $P_c \cup P_{\bar{c}}$ must correctly evaluate the clauses. By using similar arguments, it can be shown that players in $P_t \cup \{E\}$ must act as AND-gates on the truth values coming from their children.

Then, to conclude the proof, it is sufficient to observe that $E$ is correctly evaluating in $\mathbf{x}$ the truth-value of the assignment $\sigma$. Indeed, if $\sigma$ is a satisfying assignment, then all the players in $P_c$ play $T$ in $\mathbf{x}$ with probability 1, and given that all the players in $P_t \cup \{E\}$ act as AND-gates in $\mathbf{x}$, it follows that $E$ plays $T$ with probability 1 in $\mathbf{x}$ as well. On the other hand, if $\sigma$ is not a satisfying assignment, then there exists a clause in $P_c$ playing $F$ (i.e., $T$ with probability 0). Then, by construction of $\mathbf{x}$, $E$ has to play $T$ with probability 0 as well.

Exploiting these properties, we can prove the lemma as follows:

($\Rightarrow$) Let $\sigma$ be a satisfying assignment for $\Phi$, and let $\mathbf{x}$ be the Nash equilibrium such that $\sigma = \sigma^{\mathbf{x}}$, as in Property B above. The result follows because $E$ must play $T$ in $\mathbf{x}$, according to Property C.

($\Leftarrow$) It can be shown that, for any Nash equilibrium $\mathbf{x}$ for $\mathcal{G}(\Phi)$ where $E$ plays $T$, there exists a satisfying truth-value assignment for $\Phi$. Indeed, from Property A above, each player in $P_v$ plays in a deterministic way in $\mathbf{x}$. We show that $\sigma^{\mathbf{x}}$ is a satisfying assignment. Observe that each player $t \in P_t \cup \{E\}$ correctly acts as an AND-gate in $\mathbf{x}$. For the sake of contradiction, assume that there exists a player $t \in P_t \cup \{E\}$ that plays $T$ but there exists a player in $Neigh(t)$ playing $F$ in $\mathbf{x}$. Then, $t$ gets payoff 0 and gets an incentive to deviate by playing $F$, which is impossible since $\mathbf{x}$ is an equilibrium. It follows that $E$ plays $T$ if and only if all the players in $P_c$ play $T$ with probability 1. By exploiting similar arguments as above, it can be shown that players in $P_c$ are correctly evaluating all clauses. Therefore, $\sigma^{\mathbf{x}}$ is a satisfying truth-value assignment for $\Phi$. $\square$

Armed with the above construction, we can now show that deciding the existence of Nash equilibria in weakly constrained games is an NP-hard problem.

**Theorem 3.4.** *Deciding whether a weakly constrained graphical game (WCG) $\mathcal{G}$ has constrained Nash equilibria is* NP*-hard, even if:* (i) *there is only one constraint on the payoffs,* (ii) *each player is allowed to play two actions at most, and* (iii) *each player has two neighbors at most.*

**Proof.** The reduction is from 3SAT. Let $\Phi$ be a Boolean formula in conjunctive normal form, and $\mathcal{G}(\Phi)$ its associated game. Beforehand, recall that $\mathcal{G}(\Phi)$ is such that each player has two available actions, that are $T$ and $F$, and has two neighbors at most. Then, it can be shown that, adding one constraint on the payoffs, deciding the existence of a constrained Nash equilibrium for $\mathcal{G}(\Phi)$ amounts to deciding the satisfiability of $\Phi$.

Indeed, it is sufficient to observe that by virtue of the construction in Lemma 3.3, $\Phi$ is satisfiable $\Leftrightarrow$ $\mathcal{G}(\Phi)$ admits a Nash equilibrium satisfying the constraint $[pay_E = 2]$. $\square$

For completeness, note that the above proof may easily be modified in order to deal with any desired constraint payoff. Indeed, by modifying $\mathcal{G}(\Phi)$ so that player $E$ gets some payoff $\beta > 0$ in rule **(E-i)** and $\frac{\beta}{2}$ in rule **(E-ii)**, the above hardness result generalizes over Nash equilibria satisfying the constraint $[pay_E = \beta]$.

We conclude this section by stressing that the above intractability result is established by means of a construction over a game whose neighborhood relationship is not acyclic, given that each player of the form $x'_i$ depends on $x''_i$, and vice-versa. In fact, over 3-player games whose neighborhood relationship is not acyclic, it has been shown that exactly computing Nash equilibria satisfying certain constraints is algebraically infeasible, for it requires strategies over irrational numbers [16]. However, the hardness result of Theorem 3.4 is not just the intriguing computational counterpart of the algebraic viewpoint discussed in [16]. Indeed, by inspecting the proof of Theorem 3.4, it is immediate to check that constrained equilibria in $\mathcal{G}(\Phi)$ do not involve irrational numbers; hence, our result evidences, in particular, that constraints act as a source of intractability that is independent of any algebraic issue.

## 4. Hardness results for objective functions

In this section, we complete the picture of the hardness of dealing with constrained Nash equilibria, by considering the case where one would like to compute a Nash equilibrium optimizing some given objective function. As in Section 3, the analysis will be parameterized with respect to the kinds of evaluation functions on top of which objective constraints can be built.

We start observing that the games used in the reductions in the proofs of Theorems 3.1 and 3.4 have one equality constraint each, and that both these constraints prescribe that an evaluation function outputs at the desired equilibria its maximum possible value. Then, let $\mathcal{G}$ be one of these games and let $\mathcal{G}'$ be the same game as $\mathcal{G}$, but where we have to maximize the value of such an evaluation function, say $f^c$, instead of having the equality constraint. Clearly, $\mathcal{G}'$ is as hard as the original problem, because $\mathcal{G}$ has a constrained equilibria if and only if the optimal value of $f^c$ is equal to its maximum possible value. Then, computing such a value in fact solves the original constrained problem. The two following results are thus immediate consequences of the hardness of the problems shown in the proofs of Theorems 3.1 and 3.4.

**Theorem 4.1.** *Let $\mathcal{G}$ be an arbitrarily constrained graphical game. Then, computing any Nash equilibrium that optimizes $obj(\mathcal{G})$ is* NP*-hard, even if:* (i) $|constr(\mathcal{G})| = 0$, (ii) *each player is allowed to play two actions at most, and* (iii) *each player has no neighbors.*

**Theorem 4.2.** *Let $\mathcal{G}$ be a weakly constrained graphical game. Then, computing any Nash equilibrium that optimizes $obj(\mathcal{G})$ is* NP*-hard, even if:* (i) $|constr(\mathcal{G})| = 0$, (ii) *each player is allowed to play two actions at most, and* (iii) *each player has two neighbors at most.*

In fact, Theorem 4.2 immediately entails the hardness of optimizing linearly and polynomially constrained games, when each player has two neighbors at most and no hard constraint. By exploiting two hard constraints, we can instead prove that hardness holds even if each player has one neighbor at most.

**Theorem 4.3.** *Let $\mathcal{G}$ be a linearly (and, hence, polynomially) constrained graphical game. Then, computing any Nash equilibrium that optimizes $obj(\mathcal{G})$ is* NP*-hard, even if:* (i) $|constr(\mathcal{G})| = 2$, (ii) *each player is allowed to play two actions at most,* (iii) *each player has one neighbor at most, and* (iv) *the neighborhood relationship in $\mathcal{G}$ is an acyclic graph.*

**Proof.** Let us modify the construction of the game $\mathcal{G}_{ec}$ in the proof of Theorem 3.2, by removing $c_1$ (while keeping $c_2$ and $c_3$), and by adding the objective function $[\max f^{c_1}]$. Let $\mathcal{G}'_{ec}$ be the resulting game. From the proof of Claim 3.2.1, recall the expression

$$f^{c_2}(\mathbf{x}) = \sum_{i=1}^{3n}\left(\sum_{p_j^k | j=i} pay_{p_j^k}(\mathbf{x})\right) = 3 \times |\{k \mid p_{IN}^k(\mathbf{x}) = 1\}| + \sum_{q \in \bar{\mathcal{Q}}} pay_q(\mathbf{x}),$$

where $\bar{\mathcal{Q}}$ is the set of players $\{p_\ell^k \mid p_{IN}^k(\mathbf{x}) < 1\}$. If a Nash equilibrium $\mathbf{x}$ satisfies $c_2 : [f^{c_2} = 3 \times n]$, the above relationships immediately entail that $|\{k \mid p_{IN}^k(\mathbf{x}) = 1\}| \leq n$. Moreover, we claim that $f^{c_1}(\mathbf{x}) = \sum_{k=1}^m pay_{s^k}(\mathbf{x}) \leq n$. Indeed, in the case where $\sum_{k=1}^m pay_{s^k}(\mathbf{x}) = \sum_{k=1}^m r_{IN}^k(\mathbf{x}) > n$, we may conclude that $n < |\{k \mid r_{IN}^k(\mathbf{x}) > 0\}| \leq |\{k \mid s_{IN}^k(\mathbf{x}) = 1\}| \leq |\{k \mid p_{IN}^k(\mathbf{x}) = 1\}|$, which is impossible. Thus, the maximum possible output value in $\mathcal{G}'_{ec}$ of $f^{c_1}$ is $n$.

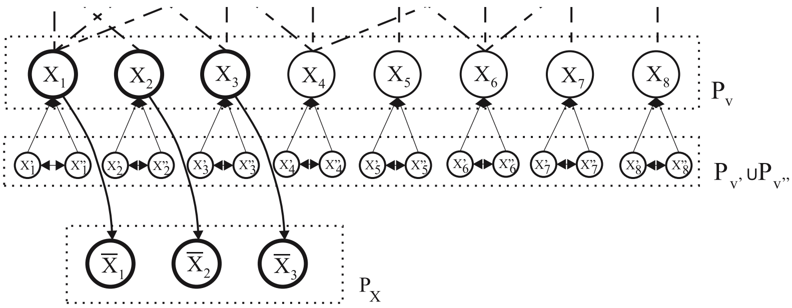



**Fig. 7.** Reduction from $X$-MAXIMAL MODEL.

Let $\mathbf{x}^*$ be any optimal Nash equilibrium of $\mathcal{G}'_{ec}$, that is, any Nash equilibrium that maximizes $f^{c_1}$ and satisfies the constraints $c_2$ and $c_3$. Recall that the constraint $c_1$ of $\mathcal{G}_{ec}$ prescribes that $f^{c_1}$ is equal to its maximum possible value $n$. Therefore, $\mathcal{G}_{ec}$ has a constrained Nash equilibrium (satisfying $c_1$, $c_2$, and $c_3$) if and only if $f^{c_1}(\mathbf{x}^*) = n$. In particular, if the latter condition holds, then $\mathbf{x}^*$ is a constrained Nash equilibrium for $\mathcal{G}_{ec}$. It follows that computing any optimal equilibrium for $\mathcal{G}'_{ec}$ is as hard as deciding whether there exists a constrained Nash equilibrium form $\mathcal{G}_{ec}$. $\square$

### *4.1. A closer look to LCGs: Optimizing the social optimum*

Among the various possible objectives that can be built on top of linear evaluation functions, a prominent role has been played in the literature by the optimization of the *social optimum*, i.e., of the function $f^{so}$ such that $f^{so}(\mathbf{x}) = \Sigma_{p\in P} pay_p(\mathbf{x})$, for any global strategy $\mathbf{x}$ over players in $P$. In this section, we focus on this function and we further elaborate the NP-hardness result of Theorem 3.2, by means of a finer analysis in terms of functional complexity classes rather than decision ones.

Recall that an NP *metric Turing machine MT* is a polynomial-time non-deterministic Turing machine that, on every computation branch, halts with the encoding of a binary number on its output tape. The output of *MT* is the maximum output number over its computations. The class OptP contains all integer functions that are computable by an NP metric Turing machine. Moreover, OptP$[O(\log n)]$ is the subclass of OptP containing all functions $f$ whose value $f(x)$ is representable with $O(\log n)$ bits, where $n = |x|$ (see [9]). For instance, computing the cardinality of a maximum clique or of a minimum vertex cover in a graph are OptP$[O(\log n)]$ functions. Then, FNP//OptP (resp., FNP//OptP$[O(\log n)]$) is the class of all partial multi-valued functions $g$ computed by polynomial-time non-deterministic Turing machines $T$ such that, for every $x$, $g(x) = T(x \cdot h(x))$, where $\cdot$ denotes the concatenation operator, and $h$ is a function in OptP (resp., in OptP$[O(\log n)]$).

Completeness results for such classes are obtained by using the notion of metric reduction: A problem $\Pi$ reduces to a problem $\Pi'$, if there are polynomial-time computable functions $g_i(x)$ and $g_s(x, y)$ mapping instances and solutions between the two problems, such that: (i) for any instance $I$ of $\Pi$, $g_i(I)$ is an instance of $\Pi'$, and $g_i(I)$ has solutions if and only if $I$ has solutions, and (ii) for any arbitrary solution $S$ of $g_i(I)$, $g_s(I, S)$ is a solution of $I$.

We can now characterize the intrinsic complexity of computing the social optimum.

**Theorem 4.4.** *Let $\mathcal{G}$ be a linearly constrained graphical game such that $obj(\mathcal{G}) = [\max f^{so}]$. Then, computing any Nash equilibrium that optimizes $obj(\mathcal{G})$ is* FNP//OptP$[O(\log n)]$*-hard, even if:* (i) $|constr(\mathcal{G})| = 0$, (ii) *each player is allowed to play two actions at most, and* (iii) *each player has two neighbors at most.*

**Proof.** Consider the FNP//OptP$[O(\log n)]$-complete problem $X$-MAXIMAL MODEL [9]: Given a formula $\Phi$ in conjunctive normal form on variables $\{X_1, \ldots, X_n\}$ and a subset $X$ of such set of variables, compute a satisfying truth-value assignment $M$ for $\Phi$ whose $X$-part is maximal, i.e., for every other satisfying assignment $M'$ that differs from $M$ on some variable in $X$, there exists a variable in $X$ which is true in $M$ and false in $M'$. Without loss of generality, we assume $\Phi$ is a satisfiable formula (but not a tautology).

The mapping $g_i$ is a polynomial time algorithm that, given such a formula $\Phi$, computes a game $\mathcal{G}^*(\Phi)$ equipped with an objective function $obj(\mathcal{G}^*(\Phi))$ such that: (i) there is a one-to-one correspondence between Nash equilibria of $\mathcal{G}^*(\Phi)$ and truth-value assignments of $\Phi$, and (ii) each Nash equilibrium optimal w.r.t. $obj(\mathcal{G}^*(\Phi))$ corresponds to an $X$-MAXIMAL MODEL.

The game $\mathcal{G}^*(\Phi)$ is a slight modification of the game $\mathcal{G}(\Phi)$ associated with the formula $\Phi$, as described in the proofs of Theorem 3.4. Besides the players in $\mathcal{G}(\Phi)$—denote the set of these players by $P$—, the set of players $P^*$ of the new game contains a fresh set of players $P_X$ corresponding to the $X$ variables. In particular, each player in $P_X$ is of the form $\bar{x}_j$, where $X_j$ is a variable in the set $X$. Player $\bar{x}_j$ may choose an action in $\{T, F\}$ and her payoff depends only on the choice of the player $x_j$ (corresponding to the variable $X_j$). Fig. 7 shows (a portion of) the neighborhood relationship for $\mathcal{G}^*(\bar{\Phi})$, where $\bar{\Phi}$ is the formula presented in Section 3.3.1 and where the set $X = \{X_1, X_2, X_3\}$.

Let $\mathbf{y}$ be a global pure strategy. Then, for each player $\bar{x}_j$ in $P_X$, her utility function $u_{\bar{x}_j}$ is such that:

**(X-i)** $pay_{\bar{x}_j}(\mathbf{y}) = 2 \times |P| + 1$, if both $\bar{x}_j$ and $x_j$ play $T$;
**(X-ii)** $pay_{\bar{x}_j}(\mathbf{y}) = 0$, in all the other cases.

Utility functions for all the other players remain unchanged, but for player $E$ that gets $2 \times |P| + (2 \times |P| + 1) \times |X| + 1$ in **(E-i)** rather than 2.

Note that, given the above utility functions, equilibria of $\mathcal{G}(\Phi)$ are "preserved" in $\mathcal{G}^*(\Phi)$, since players in $P_X \cup \{E\}$ do not affect the decision of any other player.

Consider, now, the objective [max $f^{so}$]. Given that the payoff in **(E-i)** is greater than the sum of all the possible payoffs for all the other players in $\mathcal{G}^*(\Phi)$, and given that $\Phi$ is satisfiable, any Nash equilibrium that maximizes $f^{so}$ encodes a satisfying truth-value assignment for $\Phi$, in the light of Lemma 3.3. More precisely, $\Phi$ is satisfied by the assignment encoded by the strategies of players in $P_v$ in the profile.

Moreover, because of the rule **(X-i)** and since each player in $P$ gets payoff 2 at most, it is also the case that any Nash equilibrium that maximizes $f^{so}$ also maximizes the number of players in $P_X$ choosing $T$, among all possible profiles encoding satisfying assignments for $\Phi$.

Eventually, consider the function $g_s$ that, given a formula $\Phi$ and an optimal Nash equilibrium $\mathbf{x}^*$ of $\mathcal{G}^*(\Phi)$ w.r.t. [max $f^{so}$], computes the satisfying truth-value assignment associated with $\mathbf{x}^*$. Clearly, $g_s$ can be evaluated in polynomial time and maps solutions of the game problem (optimal profiles) to solutions of $X$-MAXIMAL MODEL. Indeed, from the above reasoning, $\mathbf{x}^*$ encodes a satisfying assignment for $\Phi$ with the maximum possible number of variables in $X$ set to *true*, which is of course an $X$-maximal model of $\Phi$. □

### *4.2. Multi-objective optimization: The case of Pareto equilibria*

All the complexity results we have discussed in this section refer to the formal framework illustrated in Section 2, where there is a unique objective function to be optimized. Next, we discuss, instead, a slight extension of this framework where several objective functions have to be optimized at the same time, provided some suitable mechanisms for their combination.

In particular, even though investigating this kind of extension in its details is outside the scope of this paper, we nonetheless find it useful to illustrate some complexity results pertaining to the *Pareto equilibrium* [2]. This notion is based on the Pareto mechanism for combining individual utility functions, and it has been introduced for avoiding defection of the entire group of players in addition to that of individuals. In fact, the Pareto equilibrium is usually considered as a prototypical kind of constrained equilibrium and, hence, its investigation appears appropriate here. Actually, in the literature there are two notions of Pareto optimality leading to different notions of Pareto Nash equilibria, both studied in this section.

**Definition 4.5** (*Pareto Equilibria*). Let $\mathcal{G} = \langle P, Neigh, Act, U\rangle$ be a game and $\mathbf{x}$ be a Nash equilibrium for $\mathcal{G}$. Then,

- $\mathbf{x}$ is a Pareto equilibrium (also called *strong* Pareto equilibrium) if there does not exist a Nash equilibrium $\mathbf{y}$ for $\mathcal{G}$ such that $\forall p \in P,\ pay_p(\mathbf{x}) \leq pay_p(\mathbf{y})$, and $\exists q \in P,\ pay_q(\mathbf{x}) < pay_q(\mathbf{y})$;
- $\mathbf{x}$ is a *weak* Pareto equilibrium if there is no Nash equilibrium $\mathbf{y}$ for $\mathcal{G}$ such that, $\forall p \in P, pay_p(\mathbf{x}) < pay_p(\mathbf{y})$. □

Clearly enough, any (strong) Pareto equilibrium is also a weak Pareto equilibrium. Also, note that, if a game has a Nash equilibrium, then it has a strong – and hence a weak – Pareto Nash equilibrium, too. Therefore, the problem that attracted more attention in the literature is to check whether a given profile is indeed a Pareto equilibrium. In fact, it has been shown in [26] that this problem is co-NP-complete when pure strategies in graphical games are considered, under the notion of weak Pareto optimality. Next, we state its complexity in the setting of graphical games under mixed strategies, considering both strong and weak Pareto equilibria.

**Theorem 4.6.** *Let $\mathcal{G}$ be a graphical game, and $\mathbf{x}$ be a profile. Then, checking whether $\mathbf{x}$ is a (strong/weak) Pareto equilibrium is* co-NP*-hard, even if:* (i) $|constr(\mathcal{G})| = 0$, (ii) *each player is allowed to play two actions at most, and* (iii) *each player has three neighbors at most.*

**Proof.** Let $\Phi$ be a Boolean formula in conjunctive normal form and recall that deciding whether $\Phi$ is not satisfiable is a co-NP-hard problem [23]. W.l.o.g., assume that the assignment where all the variables evaluate to false is not satisfying.

Then, consider again the game $\mathcal{G}(\Phi) = \langle P, Neigh, Act, U\rangle$ constructed in Section 3.3.1, and for some positive value $\gamma$, let $\mathcal{G}'_\gamma = \langle P, Neigh', Act, U'\rangle$ be a new game such that: $Neigh'(p) = Neigh(p) \cup \{E\}$, for each player $p \in P - \{E\}$, and $U' = \{u_E\} \cup \{u'_p \mid p \neq E,\ u_p \in U\}$ where, for each combined strategy $\mathbf{x}$ for $\{p\} \cup Neigh'(p)$:

- $u'_p(\mathbf{x}) = \gamma u_p(\mathbf{x})$, if $E$ plays $T$ in $\mathbf{x}$, and
- $u'_p(\mathbf{x}) = u_p(\mathbf{x})$ if $E$ plays $F$ in $\mathbf{x}$.

Note that $\mathcal{G}'_\gamma$ is obtained by modifying the construction in Section 3.3.1, in such a way that each player depends on $E$, as it also appears from Fig. 8. In particular, it is relevant to note that each player but $E$ has in $\mathcal{G}'_\gamma$ one more neighbor than in $\mathcal{G}(\Phi)$. Nevertheless, it can be shown that, for any value of $\gamma \geq 1$, Nash equilibria of $\mathcal{G}(\Phi)$ are preserved in $\mathcal{G}'_\gamma$.

**Property D.** *Let $\mathbf{x}$ be a global strategy for $\mathcal{G}(\Phi)$ and $\mathcal{G}'_\gamma$ be the game constructed from $\mathcal{G}(\Phi)$, for some $\gamma \geq 1$. Then, $\mathbf{x}$ is a Nash equilibrium for $\mathcal{G}(\Phi) \Leftrightarrow \mathbf{x}$ is a Nash equilibrium for $\mathcal{G}'_\gamma$.*

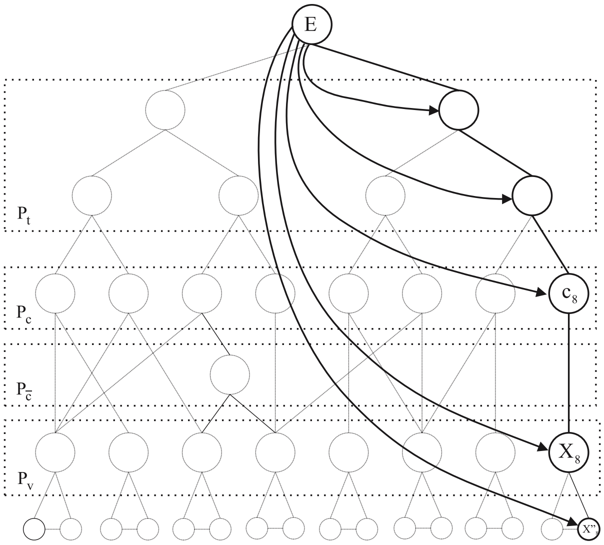



**Fig. 8.** (Part of) the neighborhood relationships in the proof of Theorem 4.6: Dependencies from $E$ are depicted for the rightmost players only.

Let us compute the expected payoff of each player $p$ in $\mathcal{G}'_{\gamma}$, denoted by $pay'_p$. By exploiting the definition of the utility functions in $U'$, it can be easily derived that $pay'_p(\mathbf{x}) = E_T \gamma pay_p(\mathbf{x}) + (1 - E_T)pay_p(\mathbf{x}) = pay_p(\mathbf{x})(1 + (\gamma - 1)E_T)$. Then, for $\gamma \geq 1$, the actual value of $E_T$ has no influence on the selection of the individual strategies, since the resulting payoff $pay'_p$ will have the same sign of the original payoff $pay_p$. It follows easily that Nash equilibria are preserved in the modified game.

In the following, let us consider the case of $\gamma > 2$. Also, let us consider the truth-value assignment $\sigma$ (which is not satisfying) where all the variables evaluate to false, and let $\mathbf{x}$ be the pure Nash equilibrium associated to $\sigma$, as it has been constructed by exploiting Property B of Lemma 3.3.

For the sake of convenience, we recall here that $\mathbf{x}$ is such that: each player in $P_v$ chooses its individual strategy according to $\sigma^{\mathbf{x}} = \sigma$; each pair of players of the form $x'_i$ and $x''_i$ choose strategy $s_2$ (see Property A); each player in $P_c \cup P_{\bar{c}}$ applies rule **(C-i)**, i.e., she correctly evaluates the clause; all the players in $P_t \cup \{E\}$, according to the rules **(T-i)** and **(E-ii)**, act as AND-gates on the inputs of their children. Notice, in particular, that $E$ has to apply rule **(E-ii)**, because $\sigma^{\mathbf{x}} = \sigma$ is not a satisfying assignment, and therefore she plays $T$ with probability 0, and she gets payoff 1 whereas its maximum available payoff is 2. Instead, we notice that all the other players get in $\mathbf{x}$ the maximum available payoff (i.e., 1) they might achieve when restricted to play over scenarios where $E$ plays $T$ with probability 0; in particular, each player but those in $P_v \cup \{E\}$ gets payoff 1, whereas players in $P_v$ get payoff 2 for they play $F$ in $\mathbf{x}$ (see, again, Property A).

Let us now show that $\mathbf{x}$ *is a (strong/weak) Pareto Nash equilibrium* $\Leftrightarrow$ $\Phi$ *is not satisfiable*. In fact, by recalling that $\mathbf{x}$ *strong* $\Rightarrow$ $\mathbf{x}$ *weak*, it suffices to show that $\mathbf{x}$ *weak* $\Rightarrow$ $\Phi$ *is not satisfiable*, and that, conversely, $\mathbf{x}$ *strong* $\Leftarrow$ $\Phi$ *is not satisfiable*.

($\Rightarrow$) Assume that $\mathbf{x}$ is a weak Pareto Nash equilibrium and, for the sake of contradiction, that $\Phi$ is satisfiable. Then, take one satisfying assignment, say $\sigma^*$, and consider the equilibrium $\mathbf{x}^*$ that is associated to $\sigma^*$ according to Property B in Lemma 3.3 and preserved by Property D above. Since $E$ plays $T$ in $\mathbf{x}^*$ with probability 1 by Property C in the lemma, we can show that each player in $\mathcal{G}'_{\gamma}$ will increase her payoff in $\mathbf{x}^*$ w.r.t. the payoff she gets in $\mathbf{x}$ (where $E$ plays $T$ with probability 0). In particular, each player in $\mathcal{G}'_{\gamma}$ but $E$ and those in $P_v$ playing $T$ will receive $\gamma$ in $\mathbf{x}^*$, whereas each player in $P_v$ playing $F$ will receive $2 \times \gamma$—see the form of the payoffs in Property D. Thus, these players get in $\mathbf{x}^*$ at least $\gamma > 2$, whereas their maximum payoff achieved in $\mathbf{x}$ is 2. Hence, they get an incentive to deviate to $\mathbf{x}^*$. Eventually, player $E$ gets also a better payoff since the formula is satisfied in $\mathbf{x}^*$ and since she can apply **(E-i)** rather than **(E-ii)**. It follows that each player gets a higher payoff if all of them jointly deviate from $\mathbf{x}$ to $\mathbf{x}^*$. Thus, $\mathbf{x}$ is not a weak Pareto Nash equilibrium. Contradiction.

($\Leftarrow$) Assume that $\Phi$ is not satisfiable and, for the sake of contradiction, that $\mathbf{x}$ is not a strong Pareto equilibrium. Let $\mathbf{x}^*$ be a Nash equilibrium witnessing that $\mathbf{x}$ is not strong Pareto. Since both $\mathbf{x}$ and $\mathbf{x}^*$ are Nash equilibria, because of Property A in Lemma 3.3, each player in $P_v$ deterministically play either $T$ or $F$ in these strategies. Therefore, $\mathbf{x}$ and $\mathbf{x}^*$ encode two truth-value assignments that are denoted by $\sigma^{\mathbf{x}}$ and $\sigma^{\mathbf{x}^*}$, respectively. By Property C in the same lemma, $E$ plays $T$ with probability 0 in $\mathbf{x}$ and gets payoff 1, because $\sigma^{\mathbf{x}}$ has to be a an assignment which is not satisfying. However, each player in $\mathbf{x}$ gets the maximum available payoff they might achieve when restricted to play over scenarios where $E$ plays $T$ with probability 0. Therefore, since $\mathbf{x}$ is not strong Pareto, it must be assumed that $E$ plays $T$ with probability greater than 0 in $\mathbf{x}^*$, to get a value higher than its value at $\mathbf{x}$. However, because of Property C again, this entails that $\sigma^{\mathbf{x}^*}$ is a satisfying assignment, which is impossible. □

As a further observation, by inspecting the proof of the above result, we may note that if the formula $\Phi$ is satisfiable, then player $E$ must play $T$ at any Pareto Nash equilibrium. Otherwise, i.e., if $\Phi$ is not satisfiable, then $E$ must play $F$ at any Nash

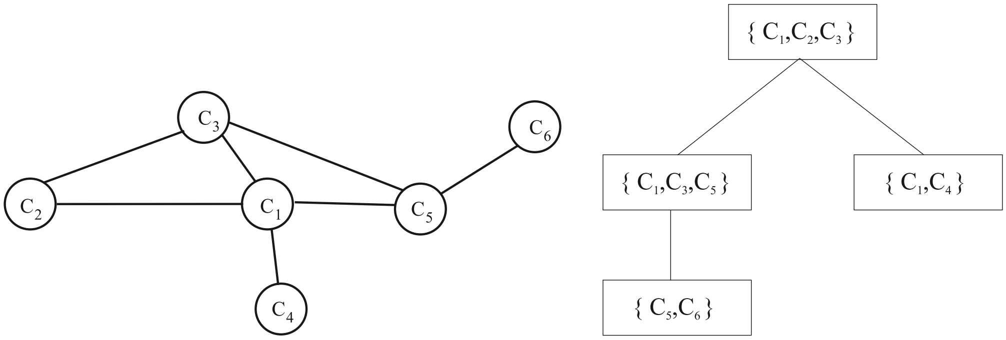



**Fig. 9.** The dependency graph for $\mathcal{G}_c$, and a tree decomposition of it.

equilibrium (and, therefore, in any Pareto equilibrium as well). Thus, the following hardness result immediately follows, which does not directly stem from the hardness results earlier discussed in the paper.

**Theorem 4.7.** *Let $\mathcal{G}$ be a graphical game. Then, computing a (strong/weak) Pareto equilibrium is both* NP-*hard and* co-NP-*hard, even if:* (i) $|constr(\mathcal{G})| = 0$, (ii) *each player is allowed to play two actions at most, and* (iii) *each player has three neighbors at most.*

## 5. Constrained Nash equilibria over pure strategies

In this section, pure strategies are considered, i.e., it is assumed that each player has to deterministically select the action to perform; and, accordingly, we shall look for Nash equilibria as those global pure strategies from which no player has an incentive to unilaterally deviate, by selecting a different action. Actually, since the existence of a mixed strategy for a player that increases her expected payoff (w.r.t. her current strategy) implies the existence of a pure strategy that does so (see, e.g., [44]), the setting we shall consider precisely coincides with that of looking for *pure Nash equilibria* over games where randomization is allowed as usual. In particular, requiring Nash equilibria to be pure acts as a source of complexity in the same manner as issuing constraints of the kinds we studied in the previous sections.

In fact, complexity issues related to pure Nash equilibria in graphical games have been investigated in [26,1,18]. In particular, it is proven in [26] that determining whether a game has a pure Nash Equilibrium is NP-hard, even in very restrictive settings (and, in fact, in absence of any further constraint). In the same paper, however, some tractable classes of games have been identified, and efficient algorithms for the computation of their pure Nash equilibria have been proposed. Yet, it was not known what happens if further constraints are issued on the game. This section precisely faces this research question.

### 5.1. Intricacy and constraints

By looking at the proofs of the theorems presented so far, one can identify two independent sources of complexity that make the problems of deciding the existence and of computing a Nash equilibrium hard. The first one is the intricacy of players interactions, as in the general case these problems are NP-hard even in the presence of very simple requirements on equilibria. The second lies in the nature of constraints, which may add additional intricacy to the game. Indeed, if arbitrary evaluation functions are considered, then the considered problems remain NP-hard even without complex players interactions.

In order to identify classes of tractable constrained games, i.e., games for which computing constrained pure Nash equilibria is feasible in polynomial time, it is therefore of utmost importance to find some good trade-off between the two factors above, and to single out "easy" scenarios with interactions and constraints that may occur in practice. In addition, it is also relevant to bound the computational requirements for evaluation functions, so that they will not represent an overhead when computing constrained equilibria. In the rest of this section, we shall move towards these three directions.

**Limiting the intricacy.** It is well-known that a fundamental structural property of graphs is *acyclicity*. Indeed, many hard problems emerging in areas such as constraint satisfaction and database query evaluation (e.g., [3,28,27,55]) turn out to be easy for acyclic structures. As commonly done in the literature on graphical games, the structure of a game $\mathcal{G}$ is represented by its *dependency graph* $\mathrm{G}(\mathcal{G}) = (P, E)$, whose vertices in $P$ coincide with the players of $\mathcal{G}$, and where there is an edge in $\{i, j\} \in E$ if $j$ is a neighbor of $i$, i.e., $j \in Neigh(i)$. For instance, Fig. 9 shows on the left the dependency graph of the game $\mathcal{G}_c$ introduced in Example 1.1.

Note that this graph encodes, in an undirected manner, the neighborhood relationship, and hence it does not take into account the fact that payoffs of a player $j$ may (directly) depend on payoffs of a player $i$ and not vice-versa. In fact, directed dependency graphs have also been considered by some authors (see, e.g., [53,54,25,13]). However, it is known (see, e.g., the discussion in [26]) and easy to see that considering the specific role of the players in the relationship of neighborhood does not help in identifying classes of structurally tractable games (but for very trivial cases). Intuitively, the choice of such a player $i$ may well depend on the choice of player $j$, as far as the possibility of reaching an equilibrium is concerned. For instance, suppose that for some choice of $i$ no choice of $j$ leads to an equilibrium, while for some choice of $i$ there are good choices of $j$. Then, only strategies of $i$ that take into account the possible strategies of $j$ may lead to equilibria.

Looking for acyclic dependency graphs appears the first promising approach for isolating tractable classes of games. However, in many practical contexts, graphs are in fact cyclic, even though often not very intricate. In these cases, it can be useful to consider some generalizations of graph acyclicity, which allow us to identify structures having some nice properties, similar to those exhibited by acyclic graphs. In particular, the notion of *treewidth* [50] will be used, which provides a measure of the degree of cyclicity of graphs, and which is currently the broadest-known (tractable) generalization of graph acyclicity.

**Definition 5.1** (*[50]*). A *tree decomposition* of a graph $G = (V, E)$ is a pair $\langle T, \chi\rangle$, where $T = (N, F)$ is a tree, and $\chi$ is a labelling function assigning to each vertex $p \in N$ a set of vertices $\chi(p) \subseteq V$, such that the following conditions are satisfied:

(1) for each vertex $b$ of $G$, there exists $p \in N$ such that $b \in \chi(p)$;
(2) for each edge $\{b, d\} \in E$, there exists $p \in N$ such that $\{b, d\} \subseteq \chi(p)$;
(3) for each vertex $b$ of $G$, the set $\{p \in N \mid b \in \chi(p)\}$ induces a connected subtree of $T$.

The *width* of the tree decomposition $\langle T, \chi\rangle$ is $\max_{p\in N} |\chi(p) - 1|$. The *treewidth* of $G$, denoted by $tw(G)$, is the minimum width over all its tree decompositions. □

A game $\mathcal{G}$ is said to have $k$-bounded treewidth if the treewidth of $G(\mathcal{G})$ is at most $k$. Treewidth is a true generalization of graph acyclicity, since it is well-known that $G(\mathcal{G})$ is acyclic if and only if $tw(G(\mathcal{G})) = 1$. Moreover, we note that deciding whether a game $\mathcal{G}$ has $k$-bounded treewidth is feasible in linear time, for any fixed natural number $k$, according to the results in [6].

**Example 5.2.** A tree decomposition for the dependency graph of the game discussed in Example 1.1 is reported on the right of Fig. 9. Note that the width is 2. And, in fact, the graph has some cycles. ◁

**Limiting global interactions.** In order to ensure that constraints do not alter the interactions as they appear from the dependency graph, we may think of focusing on weakly constrained games, for which evaluation functions are local. Actually, in many cases, most but not all the constraints are local. In these cases, it can be useful to consider some generalization of weakly constrained graphical games, where a game $\mathcal{G}$ may also have a few linear constraints defined in terms of evaluation functions that are linear but not local, hereinafter called linear *global* constraints (short: $constr^{glob}(\mathcal{G})$).

Formally, a game $\mathcal{G}$ is said *h-weakly constrained* if $constr(\mathcal{G}) = constr^{loc}(\mathcal{G}) \cup constr^{glob}(\mathcal{G})$, where $constr^{loc}(\mathcal{G})$ is a set of local constraints and $|constr^{glob}(\mathcal{G})| \leq h$, i.e., if there are at most $h$ linear global constraints defined over $\mathcal{G}$, plus an arbitrary number of local constraints.

**Smooth evaluation functions.** In addition to limiting the intricacy and the kinds of constraints, in order to identify some class of tractable constraints, we also need to carefully consider the computational requirements involved for computing the values of the various evaluation functions. Indeed, the fact that (linear and local) evaluation functions can be computed in polynomial time does not prevent that "large" output values can be obtained through them. In fact, to encode an output value that is exponential in the size of the game representation, denoted by $|\mathcal{G}|$ in the following, we just need polynomially many bits w.r.t. $|\mathcal{G}|$. Dealing with such large values is a source of additional complexity, which we limit here by focusing on *smooth* evaluation functions, i.e., roughly, on functions whose outputs values can be encoded with logarithmic space in the size of the game $\mathcal{G}$ (so that these values are in turn polynomially bounded w.r.t. $|\mathcal{G}|$). Formally, an evaluation function $f_{P'}$ is said *smooth* w.r.t. $\mathcal{G}$ if there is a polynomial function poly$(\cdot)$ such that, for each $\mathbf{x}$, $f_{P'}(\mathbf{x}) = O(\text{poly}(|\mathcal{G}|))$.

### 5.2. Easy constrained games

Fig. 10 shows a *non-deterministic* algorithm, called **DecideNashExistence**, that decides whether there exists a pure Nash equilibrium for an $h$-weakly constrained game $\mathcal{G}$ that satisfies all its constraints. In particular, the algorithm receives in input the game $\mathcal{G}$ together with a tree decomposition $\langle T, \chi\rangle$ for $G(\mathcal{G})$.

In a nutshell, **DecideNashExistence** is based on a recursive non-deterministic Boolean function *findNash* that, at the generic step, receives as its inputs a node $t$ of the tree decomposition, a combined strategy $\mathbf{x}$ for the set of players $\chi(t) \cup \bigcup_{p\in\chi(t)} Neigh(p)$, and a value $v_t^c$ for each global constraint $c$ of the game.[1] Intuitively, the function *findNash* has to extend the strategy $\mathbf{x}$ to all the players contained in the labellings of the children of the current node $t$, in such a way that the evaluation of each constraint $c$ in this extension equals the given value $v_t^c$.

In the MAIN part of the algorithm, for each constraint $c : [f_{P'} \text{ op } b] \in constr^{glob}(\mathcal{G})$, where op $\in \{<, >, =, \neq, \leq, \geq\}$, we non-deterministically guess a number $value(c)$ in the codomain of $f_{P'}$ such that $value(c)$ op $b$ holds. This way, in the recursive procedure *findNash*, $c$ is actually treated as an equality constraint, whose prescribed value for its evaluation function $f_{P'}$ is

[1] Note that for each node $p \in T$, $\chi(p)$ is a set of nodes of $G(\mathcal{G})$, which precisely corresponds to a set of players for $\mathcal{G}$. In fact, in the following, we shall use the terms node and player interchangeably.

**Input**: An $h$-weakly constrained game $\mathcal{G} = \langle P, Neigh, Act, U\rangle$, whose constraints are $constr^{glob}(\mathcal{G})$, a set of linear global constraints, with $|constr^{glob}(\mathcal{G})| \leq h$, and $constr^{loc}(\mathcal{G})$, a set of local constraints;
A tree decomposition $\langle T, \chi\rangle$ for $\mathrm{G}(\mathcal{G})$.
**Output**: TRUE iff there is a pure Nash equilibrium $\mathbf{x}$ s.t. $\forall c \in constr^{loc}(\mathcal{G}) \cup constr^{glob}(\mathcal{G})$, $\mathbf{x} \models c$.

```
Boolean Function findNash(t : nodeOf T, x : Strategy, {v_t^c : Q | c ∈ constr^glob(G)})
begin
   for each child s of t in T do
      guess a combined strategy y for χ(s) ∪ ⋃_{p∈χ(s)} Neigh(p);
      for each c ∈ constr^glob(G) do
         guess a value v_s^c;
      check that all the following conditions hold
         C1: x and y are "matching" strategies;
         C2: ∀i ∈ χ(s), ∄ a pure strategy y_i ∈ St(i) s.t.   u_i(y) < u_i(y_{-i}[y_i]);
         C3: y satisfies all the local constraints over players in χ(s);
         C4: findNash(s, y, {v_s^c | c ∈ constr^glob(G)});
      if this check fails then return FALSE;
   end for (* over children *)
   for each c ∈ constr^glob(G) do
      if v_t^c ≠ Σ_{i∈χ(t)} w_i^c × pay_i(x) + Σ_{s∈children(t)} ( v_s^c − Σ_{i∈χ(s)∩χ(t)} w_i^c × pay_i(x) ) then
         return FALSE;
   end for (* over constraints *)
   return TRUE;
end;

begin (* MAIN *)
   for each constraint c : [f_{P'} op b] ∈ constr^glob(G), where op ∈ {<, >, =, ≠, ≤, ≥},
      guess a number value(c) in the codomain of f_{P'} such that value(c) op b holds;
   let r be the root of T;
      guess a combined strategy x for χ(r) ∪ ⋃_{p∈χ(r)} Neigh(r);
   return findNash(r, x, {value(c) | c ∈ constr^glob(G)});
end.
```

**Fig. 10.** Algorithm **DecideNashExistence**.

just $value(c)$. Moreover, the root $r$ is selected and a strategy $\mathbf{x}$ is "guessed" for players in $\chi(r)$ and in their neighborhood. Then, we are ready for the first call to *findNash*.

Each time is called, *findNash* iteratively processes non-deterministically each child $s$ of the current vertex $t$ in $T$, by guessing a strategy $\mathbf{y}$ for $\chi(s)$ and their neighbors, as well as a value $v_s^c$ for each global constraint $c$. Again, $v_s^c$ is meant to denote the value of the evaluation function associated with $c$, when it is restricted to all players in the subtree rooted at $s$—recall here that each constraint $c$ is linear and, thus, it has the form $\Sigma_{i\in P'} w_i^c \times pay_i(\mathbf{x})$, where $P'$ is the set of players involved in this constraint, and $\mathbf{x}$ is the global strategy on which $c$ is evaluated. After the guess is performed, four conditions are checked:

- Firstly, since $\mathbf{y}$ is an attempt of extending the strategy $\mathbf{x}$, it should "match" with it on those players that belong to both their domains, i.e., each player in $\chi(s) \cap \chi(t)$ must play the same action in $\mathbf{x}$ and $\mathbf{y}$ (Condition C1). In particular, note that Condition C1 (inductively) guarantees that for each pair of nodes $v_1$ and $v_2$ such that $v_2$ is a descendant of $v_1$ in $T$, players in $\chi(v_1) \cap \chi(v_2)$ must play the same actions in the strategies guessed at $v_1$ and $v_2$, because of the connectedness condition (3) in Definition 5.1. Consider now two nodes $w_1$ and $w_2$ such that $r \notin \{w_1, w_2\}$ is their least common ancestor in $T$. The fact that players in $\chi(w_1) \cap \chi(w_2)$ play the same actions in the strategies guessed by the algorithm in $w_1$ and $w_2$ immediately derives from the above observation (about the preservation of the strategies among descendants) and the fact that $\chi(w_1) \cap \chi(w_2) \subseteq \chi(r)$ holds, again by condition (3) in Definition 5.1.
- Secondly, the payoffs of each player in $\chi(s)$ cannot be improved by changing single individual strategies. Thus, $\mathbf{y}$ represents what can be called a local pure Nash equilibrium (Condition C2).
- Thirdly, it is required that all the local constraints over some player in $\chi(s)$ are satisfied (Condition C3).
- And, finally, it must be possible to extend the strategy to the children of $s$. This is verified through a recursive call to *findNash* (Condition C4).

If all the above conditions are satisfied by the guessed values, it only remains to check that each global constraint $c$ is satisfied. To this end, a final further test is carried out to verify that the value of the evaluation function associated with $c$ when restricted over the players in $\chi(t)$ precisely equals the prescribed value $v_t^c$.

Next, we state the correctness of the algorithm.

**Theorem 5.3.** *Algorithm* **DecideNashExistence** *is correct. That is, it outputs* TRUE *if and only if there exists a pure Nash equilibrium satisfying all the constraints.*

**Proof.** Because of its non-deterministic nature, it is immediate to check that **DecideNashExistence** is correct whenever applied on a game $\mathcal{G}$ such that $constr^{glob}(\mathcal{G}) = \emptyset$ and on a tree decomposition $\langle T, \chi \rangle$ for $G(\mathcal{G})$. Indeed, Conditions C1, C2, C3, and C4 precisely prescribe that TRUE is returned if and only if a set of strategies exists, which are "local" equilibria (C2) and which satisfy the various local constraints (C3). Moreover, these strategies are matching (C1) and they span all the players of $\mathcal{G}$ (since each node of $G(\mathcal{G})$ occurs in the tree decomposition $T$ due to (1) in Definition 5.1 and because of the recursive call C4); hence, they induce a global strategy, which is a constrained Nash equilibria. In the following, let $\bar{\mathbf{x}}$ denote the global strategy that is "implicitly" computed by the algorithm as the union of all the matching strategies associated with the various vertices of the tree decomposition.

To conclude the proof, we need to check that **DecideNashExistence** remains correct whenever $constr^{glob}(\mathcal{G}) \neq \emptyset$. To this end, consider a global linear constraint $c : \sum_{i \in P'} w_i^c \times pay_i(\mathbf{x})$, and observe that we may assume w.l.o.g. that $P' = P$. Indeed, if this is not the case, we may just set $w_j^c = 0$, for each player $j \notin P'$. Moreover, recall that, at each vertex $t$ of $T$, the algorithm checks whether $v_t^c$ equals the following expression:

$$\sum_{i \in \chi(t)} w_i^c \times pay_i(\mathbf{x}) + \sum_{s \in children(t)} \left( v_s^c - \sum_{i \in \chi(s) \cap \chi(t)} w_i^c \times pay_i(\mathbf{x}) \right).$$

For each node $t$ of $T$, let $T_t$ denote the set of all the players in the labellings occurring in the subtree of $T$ rooted at $t$. Let also $f_t^c$ denote the evaluation function $f^c$ restricted over the players in $T_t$. Then, by definition of tree decomposition, it follows that $T_t = \chi(t) \cup \bigcup_s (T_s - \chi(s) \cap \chi(t))$. Hence, by linearity of the evaluation functions, it is the case that:

$$f_t^c(\bar{\mathbf{x}}) = \sum_{i \in \chi(t)} w_i^c \times pay_i(\bar{\mathbf{x}}) + \sum_{s \in children(t)} \left( f_s^c(\bar{\mathbf{x}}) - \sum_{i \in \chi(s) \cap \chi(t)} w_i^c \times pay_i(\bar{\mathbf{x}}) \right).$$

By comparing the two expressions above, we conclude that $v_t^c$ coincides with $f_t^c(\bar{\mathbf{x}})$, over each vertex $t$ of $T$, whenever the final check over $c$ in **DecideNashExistence** does not return FALSE. In particular, $v_r^c$ (where $r$ is the root of the tree decomposition) is initially set to the value prescribed as the output of the evaluation function in the constraint $c$ and, hence, FALSE is returned if and only if the constraint $c$ cannot be satisfied (provided all the other conditions are met). □

We can now show that computing a constrained pure Nash equilibrium is feasible in LOGCFL by means of the algorithm **DecideNashExistence**, if its input game belongs to a class of games having bounded treewidth and smooth constraints.

For completeness, we recall here that the class LOGCFL consists of those decision problems that are logspace reducible to a context-free language, and that $L^{LOGCFL}$ is the class of functions computed by deterministic logspace Turing transducers with LOGCFL oracles. Since LOGCFL $\subseteq AC_1 \subseteq NC_2$, problems in LOGCFL are all highly parallelizable [32,47]. Moreover, towards establishing our result, we exploit the fact that the composition of two functions computable by $L^{LOGCFL}$ transducers is itself computable in $L^{LOGCFL}$ [29].

**Theorem 5.4.** *Let $k$ and $h$ be natural numbers, and let $\mathcal{C}_{h,k}$ be a class of $h$-weakly constrained graphical games having $k$-bounded treewidth, and smooth constraints only. Then, given any game $\mathcal{G} \in \mathcal{C}_{h,k}$, deciding whether $\mathcal{G}$ has a pure Nash equilibrium satisfying all its constraints is feasible in* LOGCFL.

**Proof.** Let $\mathcal{G} = \langle P, Neigh, Act, U \rangle$ be a game in $\mathcal{C}_{h,k}$. Firstly, note that we can compute a $k$-width tree decomposition $\langle \bar{T}, \bar{\chi} \rangle$ for the graph $G(\mathcal{G})$ in $L^{LOGCFL}$, or answering "no" if the treewidth of this graph is greater than $k$ [29] (that is, if actually it does not belong to $\mathcal{C}_{h,k}$). Moreover, it is well-known that such a decomposition can be computed in a normal form (without redundancies) such that the number of vertices of $\bar{T}$ is bounded by the number of nodes in the input graph, which means by the number of players in $\mathcal{G}$, in our case. For our algorithm, it is convenient to transform $\langle \bar{T}, \bar{\chi} \rangle$ into an equivalent decomposition $\langle T, \chi \rangle$ over a binary tree $T$ as follows. Starting from the root, for each vertex $v$ of $\bar{T}$ whose children are $c_1, \ldots, c_n$, we add to $T$ the vertices $v, c_1, \ldots, c_n$ with the same labeling as in $\bar{\chi}$ plus the novel vertices $vc_1, \ldots, vc_{n-1}$. Let $v = vc_0$; then, for each $0 \leq i < n$, $vc_i$ has two children in $T$, namely $c_{i+1}$ and $vc_{i+1}$, with $\chi(vc_i) = \bar{\chi}(v)$. Note that $\langle T, \chi \rangle$ is still a $k$-width tree decomposition of $G(\mathcal{G})$, which moreover can be built by an $L^{LOGCFL}$ transducer.

Let us now consider the problem of deciding whether there is a pure Nash equilibrium satisfying all the smooth constraints of the given game. From Theorem 5.3, we know that the problem can be solved by means of Algorithm **DecideNashExistence**, with $\mathcal{G}$ and $\langle T, \chi \rangle$ as its inputs. Thus, we will next focus on the complexity of this algorithm, by using an important characterization of LOGCFL by Alternating Turing Machines. As in [52], we define a *computation tree* of an ATM $M$ on an input string $w$ as a tree whose nodes are labeled with configurations of $M$ on $w$, such that the descendants of any non-leaf labeled by a universal (existential) configuration include all (resp. one) of the successors of that configuration. A computation tree is *accepting* if the root is labeled with the initial configuration, and all the leaves are accepting configurations. Thus, an accepting tree yields a certificate that the input is accepted. A complexity measure considered by [52] for the alternating Turing machine is the tree-size, i.e., the minimal size of an accepting computation tree. A decision problem $P$ is solved by an alternating Turing machine $M$ within *simultaneous* tree-size and space bounds $Z(n)$ and $S(n)$ if, for every "yes" instance $w$ of $P$, there is at least one accepting computation tree for $M$ on $w$ of size (number of nodes) $\leq Z(n)$, each node of which represents a configuration using space $\leq S(n)$, where $n$ is the size of $w$. (Further, for any "no" instance $w$ of $P$ there is no accepting computation tree for $M$.) In fact, [52] proved that LOGCFL coincides with the class of all decision problems recognized by ATMs operating simultaneously in tree-size $O(n^{O(1)})$ and space $O(\log n)$.

By exploiting the arguments introduced in [29], we can note that **DecideNashExistence** can be implemented as a logspace alternating Turing machine $M$ with a polynomially-bounded computation tree. Indeed, each guess of **DecideNashExistence** can be implemented with existential configurations of $M$, while checks can be implemented with universal configurations. Importantly, all the information that has to be kept in each configuration of the machine can be encoded in logspace. Indeed,

(1) Each strategy for a player $p$ and her neighbors is encoded through the index of the row in the table representing $p$'s utility function;
(2) Each strategy $\mathbf{y}$ for $\chi(s) \cup \bigcup_{p\in\chi(s)} Neigh(p)$ can be encoded through the $k+1$ indices (recall that $|\chi(s)| \leq k+1$) referencing the strategy of each player in $\chi(s)$. Thus, it requires again logspace, because $k$ is a constant;
(3) For each smooth constraint $c$ among the (at most) $h$ global constraints of $\mathcal{G}$, $v_t^c$ is logspace bounded; also, for each child $s$ of $t$, $v_s^c$ is logspace bounded, as well.

In more details, note that the procedure *findNash* is invoked recursively for each child $s$ of $t$, and requires logspace cells on the worktape for storing the information associated with such a vertex $s$. However, recall that, by construction, the (modified) decomposition tree $T$ is a binary tree, and thus $t$ has at most two children and at each call of *findNash* all such values are stored in logspace. Moreover, note that the number of such calls is equal to the number of vertices in the decomposition tree $T$, and thus the tree-size of the alternating Turing machine is clearly polynomial in the input size. To be more precise, we finally observe that all deterministic checks performed in the algorithm (e.g., checking whether $\mathbf{x}$ and $\mathbf{y}$ coincide on the players they have in common, or checking whether the local constraints are satisfied) are easily done in logspace. Technically, they are implemented as further branches of universal configurations of the alternating Turing machine $M$. It is well-known that such branches lead only to a polynomial increment of the tree-size of $M$.

It follows that Algorithm **DecideNashExistence** is in LOGCFL, and the theorem follows by the results in [29], stating that LOGCFL equals $\mathrm{L}^{\mathrm{LOGCFL}}$ (and, also, its closure under $\mathrm{L}^{\mathrm{LOGCFL}}$ reductions). Indeed, this entails that our $\mathrm{L}^{\mathrm{LOGCFL}}$ preprocessing steps (computation of the tree decomposition and subsequent binarization) do not increase the complexity. □

As a final remark, let us observe that the algorithm in Fig. 10 can be used for computing a Nash equilibrium as well, by exploiting the information in the proof tree of the ATM $M$, as described in [29,28]. In fact, a Nash equilibrium can be obtained by (the encoding of) the strategies associated with the configurations of $M$.

**Corollary 1.** *Let $k$ and $h$ be natural numbers, and let $\mathcal{C}_{h,k}$ be a class of $h$-weakly constrained graphical games having $k$-bounded treewidth, and smooth constraints only. Then, given any game $\mathcal{G} \in \mathcal{C}_{h,k}$, computing a pure Nash equilibrium of $\mathcal{G}$ satisfying all its constraints (if any) is feasible in the functional version of* LOGCFL, *that is, in (functional)* $\mathrm{L}^{\mathrm{LOGCFL}}$.

### *5.3. Further tractable classes of games*

Slight modifications of **DecideNashExistence** can be used to extend the tractability frontier for constrained pure Nash equilibria. For instance, we may relax the condition that local constraints should be linear (as required by their definition), and consider any kind of constraints over single players and their neighborhood, whose associated function can be evaluated in polynomial time. It can be shown that **DecideNashExistence** is correct even for such a generalizations of $\mathcal{C}_{h,k}$, and it can still run in polynomial time, but it is not parallelizable, in general. Indeed, since local constraints are no longer required to be linear, checking whether they are satisfied may require large branches of the alternating Turing machine, leading to an exponential tree-size. It follows that such an evaluation is feasible in *Alternating LogSpace* (which coincides with polynomial-time), rather than in the lower class LOGCFL.

Also, **DecideNashExistence** can be adapted for computing a Nash equilibrium optimizing a given linear objective function (not necessarily smooth). In fact, if $f$ is the function to be optimized (say, minimized), it suffices to perform a binary search over the codomain of $f$, by calling at each step **DecideNashExistence** with an additional constraint of the form $[f \leq b]$. Since $f$ is linear, its maximum output values may be encoded with a polynomial number of bits. Thus, polynomially many steps of the binary search suffice to compute the optimum value, and hence, with a final call to (the search version of) **DecideNashExistence**, the optimal Nash equilibrium.

**Theorem 5.5.** *Let $k$ and $h$ be natural numbers, and let $\mathcal{C}_{h,k}$ be a class of $h$-weakly constrained graphical games having $k$-bounded treewidth, and smooth constraints only. Then, given any game $\mathcal{G} \in \mathcal{C}_{h,k}$, computing a constrained pure Nash equilibrium of $\mathcal{G}$ if any, which optimizes a (possibly non-smooth) linear objective function is feasible in polynomial time.*

As a final remark, we note that the techniques exploited in Algorithm **DecideNashExistence** mainly rely on the the crucial property that linear functions distribute over the various players. From this observation, a further generalization of our tractability results can be obtained. Indeed, by the same line of reasoning as in the previous proofs, it can be shown that all such results hold even if, instead of having linear functions, we have evaluation functions of the form $\oplus_{i\in P} f_i(pay_i(\mathbf{x}))$, where $\mathbf{x}$ is the global strategy on which the function is evaluated, $f_i$ is any polynomial function mapping rational numbers to rational numbers, and $\oplus$ is an associative and commutative binary operator that distributes over min and max.

## 6. Conclusion

A comprehensive study of Nash equilibria in graphical games has been provided in this paper, where one looks only for those equilibria having some desirable properties. In particular, a general framework to define constraints has been defined, and bad and good news for pure and mixed Nash equilibria have been found. It turned out that even simple attempts of constraining game outcomes immediately unsettle our only certainty. Indeed, the existence of a (constrained) mixed Nash equilibrium is no longer guaranteed, and its computation is unlikely to be tractable. However, for the case of pure strategies, it has been observed that computing a Nash equilibrium satisfying all constraints is feasible in polynomial time for games having tree-like dependency graphs, even if there is a bounded number of smooth linear global constraints and an arbitrary number of local constraints. For this case, efficient parallelizable algorithms have been discussed.

It is worthwhile noting that no membership results have been provided in this paper for the setting of mixed strategies. Indeed, Nash gave in [43] an example of a 3-player, finite-action game with a unique irrational Nash equilibrium, although all payoffs are rational numbers. Hence, it makes sense to discuss membership results only in cases where we either look for alternative symbolic representations of Nash equilibria [39], or more pragmatically consider approximate equilibria (as in, e.g., [39,34,17]). Intuitively, an $\epsilon$-equilibrium is a set of strategies such that each player cannot increase her payoff by a fixed amount ($0 < \epsilon < 1$) by unilaterally deviating to another strategy. Obviously, an $\epsilon$-equilibrium is not guaranteed to be close to a Nash equilibrium. However, some studies (see, e.g., [39,34]) show that a tuple of mixed strategies that is $\delta$-close to a Nash equilibrium is an $\epsilon$-equilibrium, for some $\epsilon$ depending on $\delta$ and on the parameters of the game. Then, a simple way for computing Nash equilibria is to consider "discretized" mixed strategies, where probabilities range over the multiples of $\delta$ rather than in the full interval [0..1]. In particular, each player $p$ is allowed to play any action, say $a$, with probability $p_a \in \{0, \delta, 2\delta, \ldots, 1\}$. Clearly enough, this is equivalent to have the player $p$ that *deterministically* selects some new actions $a_0, a_\delta, a_{2\delta}, \ldots, a_1$ getting, for each strategy $\mathbf{x}$, payoff $0, \delta pay_p(\mathbf{x}), 2\delta pay_p(\mathbf{x}), \ldots, pay_p(\mathbf{x})$, respectively. Eventually, over these discretized versions, techniques and results given in [26] for pure Nash equilibria might be exploited. Exploring this issue is an interesting avenue of further research; in particular, a suitable notion of approximate satisfaction for constraints might be given, and the problem of checking whether the results in [39] carry out over constrained games might be investigated.

Our analysis of the complexity of constrained equilibria is parameterized w.r.t. the maximum number of neighbors for the game players. Our results are often tight, in that enforcing further limitations on the number of neighbors would trivialize the problem. However, for the case of weakly constrained games where we established hardness results for the case of each player having two neighbors at most (cf. Theorem 3.4 and of Theorem 4.2), it is open whether intractability may emerge even for games where each player admits one neighbor at most. Also, it would be interesting to investigate whether intractability for linearly constrained games (proved in the paper for games with three constraints) holds even when just one constraint is considered.

We conclude by recalling that a useful mapping between strategic games in normal and graphical form has been recently described [25]. Thus, with respect to the problem of deciding the existence of Nash equilibria satisfying simple kinds of constraints, one may wonder whether some of our NP-hardness results can alternatively be established by just combining this mapping with the results in [10,11] for games in normal form. Actually, this is not the case, because the mapping is known to "preserve" the Nash equilibria of the game at hand, but it is not yet explored whether it also preserves (in some sense) the satisfaction of constraints imposed over them. In any case, by exploiting this mapping, we might, in principle, just establish NP-hardness results for graphical games where each player depends on three players at most (because this is the kind of game into which an arbitrary game in normal form can be transformed according to [25]), whereas results in this paper have been established for games where each player depends on two other players at most. Instead, having a mapping between games in graphical form and games in normal form that preserves constraints imposed on them, would be very useful for the other way around. Indeed, one would get several additional results for games in normal form as corollaries of the results presented here. Thus, exploring this issue constitutes a further interesting avenue of research.

## References

[1] A. Alvarez, C. Gabarro, M. Serna, Pure Nash equilibria in games with a large number of actions, in: Electronic Colloquium on Computational Complexity, 2005, pages Report TR05–031.
[2] R. Aumann, Accetable points in general cooperative $n$-person games, Contribution to the Theory of Games IV (1959).
[3] C. Beeri, R. Fagin, D. Maier, M. Yannakakis, On the desirability of acyclic database schemes, Journal of the ACM 30 (3) (1983) 479–513.
[4] N.A.R. Bhat, K. Leyton-Brown, Computing Nash equilibria of action-graph games, in: Proc. of the 20th Conference on Uncertainty in Artificial Intelligence, AUAI '04, AUAI Press, Arlington, VA, United States, 2004, pp. 35–42.
[5] B. Blum, C. Shelton, D. Koller, A continuation method for Nash equilibria in structured games, Journal of Artificial Intelligence Research 24 (2006) 457–502.

[6] H. Bodlaender, A linear-time algorithm for finding tree-decompositions of small treewidth, SIAM Journal on Computing 25 (1996) 1305–1317.
[7] X. Chen, X. Deng, 3-NASH is PPAD-complete, in: Electronic Colloquium on Computational Complexity, 2005, pages Report TR05–134.
[8] X. Chen, X. Deng, Settling the complexity of two-player Nash equilibrium, in: Proc. of the 47th Annual IEEE Symposium on Foundations of Computer Science, FOCS'06, IEEE Computer Society, Los Alamitos, CA, USA, 2006, pp. 261–272.
[9] Z. Chen, S. Toda, The complexity of selecting maximal solutions, Information and Computation 119 (2) (1995) 231–239.
[10] V. Conitzer, T. Sandholm, Complexity results about Nash equilibria, in: Proc. of the 18th International Joint Conference on Artificial Intelligence, IJCAI'03, Montreal, Canada, 2003, pp. 765–771.
[11] V. Conitzer, T. Sandholm, Complexity results about Nash equilibria, Games and Economic Behavior 63 (2008) 621–641.
[12] A. Czumaj, B. Vocking, Tight bounds for worst-case equilibria, in: Proc. of the 13th ACM-SIAM Symp. on Discrete Algorithms, 2002, p. 413420.
[13] C. Daskalakis, P.W. Goldberg, C.H. Papadimitriou, The complexity of computing a Nash equilibrium, in: Proc. of the 38th Annual ACM Symposium on Theory of Computing, STOC'06, ACM Press, New York, NY, USA, 2006, pp. 71–78.
[14] C. Daskalakis, C. Papadimitriou, Three-player games are hard, in: Electronic Colloquium on Computational Complexity, 2005, pages Report TR05–139.
[15] E. Elkind, L.A. Goldberg, P.W. Goldberg, Nash equilibria in graphical games on trees revisited, in: Proc. of the 7th ACM Conference on Electronic Commerce, EC'06, ACM Press, New York, NY, USA, 2006, pp. 100–109.
[16] E. Elkind, L.A. Goldberg, P.W. Goldberg, Computing good Nash equilibria in graphical games, in: Proc. of the 8th ACM Conference on Electronic Commerce, EC'07, 2007, pp. 162–171.
[17] K. Etessami, M. Yannakakis, On the complexity of nash equilibria and other fixed points, in: Proc. of the 48th Annual IEEE Symposium on Foundations of Computer Science, FOCS'07, 2007, pp. 113–123.
[18] A. Fabrikant, C. Papadimitriou, K. Talwar, The complexity of pure Nash equilibria, in: Proc. of the 36th Annual ACM Symposium on Theory of Computing, STOC'04, Chicago, IL, USA, 2004, pp. 604–612.
[19] S. Fischer, B. V ocking, On the structure and complexity of worst-case equilibria, Theoretical Computer Science 378 (2007) 165–174.
[20] D. Fotakis, S. Kontogiannis, E. Koutsoupias, M. Mavronicolas, P. 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'02, Malaga, Spain, 2002, pp. 123–134.
[21] M. Gairing, T. Lucking, M. Mavronicolas, B. Monien, P. Spirakis, Selfish routing; extreme Nash equilibria, Theoretical Computer Science 343 (2005) 133–157.
[22] Y. Gal, A. Pfeffer, Reasoning about rationality and beliefs, in: Proc. of the 3rd International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS'04, New York, NY, USA, 2004, pp. 774–781.
[23] M. Garey, D. Johnson, Computers and Intractability. A Guide to the Theory of NP-completeness, Freeman and Comp., NY, USA, 1979.
[24] I. Gilboa, E. Zemel, Nash and correlated equilibria: Some complexity considerations, Games and Economic Behaviour 1 (1989) 80–93.
[25] P.W. Goldberg, C.H. Papadimitriou, Reducibility among equilibrium problems, in: Proc. of the 38th annual ACM symposium on Theory of Computing, STOC'06, ACM Press, New York, NY, USA, 2006, pp. 61–70.
[26] G. Gottlob, G. Greco, F. Scarcello, Pure Nash equilibria: Hard and easy games, Journal of Artificial Intelligence Research 24 (2005) 357–406.
[27] G. Gottlob, N. Leone, S. Scarcello, A comparison of structural csp decomposition methods, Artificial Intelligence 124 (2) (2000) 243–282.
[28] G. Gottlob, N. Leone, S. Scarcello, The complexity of acyclic conjunctive queries, Journal of the ACM 48 (3) (2001) 431–498.
[29] G. Gottlob, N. Leone, S. Scarcello, Computing logcfl certificates, Theoretical Computer Science 270 (1–2) (2002) 761–777.
[30] G. Greco, F. Scarcello, Bounding the uncertainty of graphical games: The complexity of simple requirements, Pareto and strong Nash equilibria, in: Proc. of the 21st Conference on Uncertainty in Artificial Intelligence, UAI'05, 2005, pp. 225–232.
[31] G. Greco, S. Scarcello, Constrained pure Nash equilibria in graphical games, in: Proc. of the 16th European Conference on Artificial Intelligence, ECAI'04, Valencia, Spain, 2004, pp. 181–185.
[32] D. Johnson, A catalog of complexity classes, in: Handbook of Theoretical Computer Science, Volume A: Algorithms and Complexity, 1990, pp. 67–161.
[33] M. Kearns, M. Littman, S. Singh, An efficient exact algorithm for singly connected graphical games, in: Proc. of the 14th International Conference on Neural Information Processing Systems, NIPS'01, Vancouver, British Columbia, Canada, 2001, pp. 817–823.
[34] M. Kearns, M. Littman, S. Singh, Graphical models for game theory, in: Proc. of the 17th International Conference on Uncertainty in AI, UAI'01, Seattle, Washington, USA, 2001, pp. 253–260.
[35] M. Kearns, Y. Mansour, Efficient Nash computation in large population games with bounded influence, in: Proc. of the 18th International Conference on Uncertainty in AI, UAI'02, Edmonton, Alberta, Canada, 2002, pp. 259–266.
[36] E. Koutsoupias, M. Mavronicolas, P. Spirakis, Approximate equilibria andball fusion, Theory of Computing Systems 36 (2003) 683–693.
[37] E. Koutsoupias, C. Papadimitriou, Worst case equilibria, in: Proc. of the 16th Symposium on Theoretical Aspects of Computer Science, STACS'99, Trier, Germany, 1999, pp. 404–413.
[38] K. Leyton-Brown, M. Tennenholtz, Local-effect games, in: Proc. of the 18th International Joint Conference on Artificial Intelligence, IJCAI'03, Montreal, Canada, 2003, pp. 772–780.
[39] R. Lipton, E. Markakis, Nash equilibria via polynomial equations, in: Proc. of the 6th Latin American Symposium on Theoretical Informatics, LATIN'04, 2004, pp. 413–422.
[40] M. Mavronicolas, P. Spirakis, The price of selfish routing, in: Proc. of the 33rd Annual ACM Symposium on the Theory of Computing, STOC'01, 2001, pp. 510–519.
[41] D. Monderer, L. Shapley, Potential games, Games and Economic Behavior 14 (1996) 124–143.
[42] P.L. Mura, Game networks, in: Proc. of the 16th Annual Conference on Uncertainty in Artificial Intelligence, UAI'00, 2000, pp. 335–342.
[43] J. Nash, Non-cooperative games, Annals of Mathematics 54 (2) (1951) 286–295.
[44] M. Osborne, An Introduction to Game Theory, Oxford University Press, 2002.
[45] M. Osborne, A. Rubinstein, A Course in Game Theory, MIT Press, 1994.
[46] G. Owen, Game Theory, Academic Press, New York, 1982.
[47] C. Papadimitriou, Computational Complexity, AAddison-Wesley, Reading, Mass, 1994.
[48] C. Papadimitriou, On the complexity of the parity argument and other inefficient proofs of existence, Journal of Computer and System Sciences 48 (3) (1994) 498–532.
[49] C. Papadimitriou, Algorithms, games, and the internet, in: Proc. of the 28th International Colloqium on Automata, Languages and Programming, ICALP'01, Crete, Greece, 2001, pp. 1–3.
[50] N. Robertson, P. Seymour, Graph minors II. Algorithmic aspects of tree width, Journal of Algorithms 7 (1986) 309–322.
[51] R. Rosenthal, A class of games possessing pure-strategy Nash equilibria, International Journal of Game Theory 2 (1973) 65–67.
[52] W. Ruzzo, Tree-size bounded alternation, Journal of Cumputer and System Sciences 21 (1980) 218–235.
[53] G. Schoenebeck, S. Vadhan, The computational complexity of Nash equilibria in concisely represented games, in: Electronic Colloquium on Computational Complexity, 2005, pages Report TR05–052.
[54] G. Schoenebeck, S. Vadhan, The computational complexity of Nash equilibria in concisely represented games, in: Proc. of the 7th ACM Conference on Electronic Commerce, EC'06, ACM Press, New York, NY, USA, 2006, pp. 270–279.
[55] M. Vardi, Constraint satisfaction and database theory: A tutorial, in: Proc. of the 19th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, Dallas, Texas, USA, 2000, pp. 76–85.
[56] D. Vickrey, D. Koller, Multi-agent algorithms for solving graphical games, in: Proc. of the 18th National Conference on Artificial Intelligence, AAAI'02, Edmonton, Alberta, Canada, 2002, p. 345251.
[57] J. von Neumann, O. Morgenstern, Theory of Games and Economic Behavior, Princeton University Press, 1944.