



# Analysis and Computation of the Outcomes of Pure Nash Equilibria in Two-Player Extensive-Form Games

Paolo Zappalà<sup>1,2</sup> · Amal Benhamiche<sup>1</sup> · Matthieu Chardy<sup>1</sup> ·  
Francesco De Pellegrini<sup>2</sup> · Rosa Figueiredo<sup>2</sup>

Accepted: 1 August 2024

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2024

## Abstract

The outcomes of extensive-form games are the realisation of an exponential number of distinct strategies, which may or may not be Nash equilibria. The aim of this work is to determine whether an outcome of an extensive-form game can be the realisation of a Nash equilibrium, without recurring to the cumbersome notion of normal-form strategy. We focus on the minimal example of pure Nash equilibria in two-player extensive-form games with perfect information. We introduce a new representation of an extensive-form game as a graph of its outcomes and we provide a new lightweight algorithm to enumerate the realisations of Nash equilibria. It is the first of its kind not to use normal-form brute force. The algorithm can be easily modified to provide intermediate results, such as lower and upper bounds to the value of the utility of Nash equilibria. We compare this modified algorithm to the only existing method providing an upper bound to the utility of any outcome of a Nash equilibrium. The experiments show that our algorithm is faster by some orders of magnitude. We finally test the method to enumerate the Nash equilibria on a new instances library, that we introduce as benchmark for representing all structures and properties of two-player extensive-form games.

**Keywords** Extensive-form games · Nash equilibria · Graph algorithm · Complexity

Amal Benhamiche, Matthieu Chardy, Francesco De Pellegrini, Rosa Figueiredo have contributed equally to this work.

✉ Paolo Zappalà  
paolo.zappala@alumni.univ-avignon.fr

Amal Benhamiche  
amal.benhamiche@orange.com

Matthieu Chardy  
matthieu.chardy@orange.com

Francesco De Pellegrini  
francesco.de-pellegrini@univ-avignon.fr

Rosa Figueiredo  
rosa.figueiredo@univ-avignon.fr

<sup>1</sup> Orange Innovation, Orange, 44 Avenue de la République, 92320 Châtillon, France

<sup>2</sup> LIA, Avignon Université, Campus Jean Henri Fabre, 339 Chemin des Meinajaries, 84140 Avignon, France

# 1 Introduction

Extensive-form games represent players acting in sequence one after another (cf. [1]). These games are deployed to model situations in many domains, such as e.g., macroeconomics (cf. [2]), law (cf. [3]) and security (cf. [4]). However, from a computational standpoint, extensive-form games are not easy to deal with. Existing methods to compute solutions of a generic extensive-form game do not scale with the game size. In this work, we introduce a graph-based representation for finite extensive games with two players and perfect information. Such representation permits to resort to algorithms on graphs in the game resolution. Hence, building on what we denote *graph form*, we provide an efficient method to enumerate all the solutions of a game, i.e., to answer to the question whether or not a generic outcome of a game is a Nash equilibrium or not.

*Computing Solutions.* The standard solution of a game used in this work is the *Nash equilibrium* (cf. [5]), covering many foreseeable scenarios of game-theoretical models. The *subgame perfect equilibrium* (SPE) is the customary solution of extensive-form games (cf. [6]). A practical advantage of a SPE is that it can be computed with backward induction. On the other hand, the suitability of SPE and the possibility to employ different notions of equilibria beyond the SPE has been explored since its introduction (cf. [7]). For instance, the work [8] discusses the theoretical assumptions underlying the notion of SPE and, by restricting the horizon of acceptance of the backward induction principle acceptable by a player, it concludes that such solution does not provide complete insight into the possible equilibria. Later works, e.g., [9], show results of empirical tests on basic sample games proving that other Nash equilibria rather than the SPE are attained in practice. The discussion on refined equilibrium concepts is beyond the scope of this paper. Indeed, we are rather interested in analysing the concept of Nash equilibrium, whose fundamental role remains unquestioned. Extensive-form games, in particular, can have other non-SPE equilibria. In this work we analyse Nash equilibria independently from their properties and their corresponding refined equilibrium concepts.

To this respect, the algorithms provided in the present work offer insight into the whole set of Nash equilibria outcomes which can be ascribed to a given extensive-form game.

To the best of our knowledge, there is no efficient method to compute all the Nash equilibria of an extensive-form game. Identifying a Nash equilibrium for a generic game in normal form belongs to the class of PPAD-complete problems (cf. [10]) introduced by [11]. The most known algorithm to identify a Nash equilibrium is the Lemke-Howson algorithm (cf. [12]), that consists in identifying a completely labeled pair of vertices of two polytopes representing the game. Enumerating the Nash equilibria of a game corresponds to enumerating all the completely labeled pair of vertices of the two polytopes. Methods to solve such problem are inefficient and require a large amount of memory (cf. [13]). Recent algorithms have proven to be space efficient (cf. [14]), while others resort to parallel computing to overcome such issue (cf. [15]). In a generic extensive-form game, the number of *strategies* available to a player is typically exponential in the *size* of the game, defined as the number of its *outcomes* (cf. [16]). Transforming an extensive-form game in normal form is therefore highly inefficient, because it relies on the enumeration of an exponential number of strategies. The structure of extensive-form games can be exploited to introduce more efficient algorithms. A Mixed Integer Linear model introduced by [17] provides one Nash equilibrium and is linear in the size of the game. However, no information is provided on possible other Nash equilibria. This method proves to be highly efficient in zero-sum games (cf. [16]), but it might have exponential time complexity in a generic game. A variant of this method (cf. [18]) allows

to find the extreme Nash equilibria, which are identified by all the vertices of the polytope corresponding to a Mixed Integer Linear model. Govindan and Wilson [19] describe how to adapt homotopy-based methods, conceived for normal-form games, in order to determine an approximated Nash equilibrium for extensive-form games. However, how to control the set of equilibria returned by such procedures may not be immediate. To this respect, the graph-based enumeration proposed in this work represents, in its simplicity, a convenient algorithmic tool to provide pure Nash equilibria outcomes, even when the aim is to inspect those satisfying prescribed utility ranges. For a general introduction to homotopy-based methods, see [20].

*Outcomes of Nash Equilibria.* Since the number of strategy profiles is exponential in the size of the game, the number of Nash equilibria can also be potentially exponential. Throughout this work, the focus of the analysis is thus set on the outcomes of the Nash equilibria, rather than on the Nash equilibria themselves. This choice, which could be a trifling remark in a generic game, is relevant when dealing with extensive-form games for two reasons. First, the number of outcomes of Nash equilibria is proportional to the size of the game. Second, a large class of extensive-form games can be described in a compact way using their outcomes rather than enlisting the full game-tree. We mention, among others, the important case of timing games, the case of pricing games and the case of scheduling games. In timing games (cf. [21]), players have to perform a single action within an interval of time, having the possibility of observing the actions of the opponents once they are taken. An outcome of a timing game is just a vector of time values, one for each player. Instead, the game-tree representation of a timing game, as well as that of the following games, requires to list all the possible combinations of choices that players can perform. In a pricing game (cf. [22]), companies have to adjust the prices of their goods in order to avoid churn towards competitors and maximise their profits. The outcomes are the vectors of prices chosen in discrete intervals of time. In the scheduling game (cf. [23]), ordered players choose in sequence a machine to perform one job, being aware of the decisions taken by their predecessors; the objective of any player is to minimise the congestion of the machine she chooses. The outcome of the game is a vector of assignments of players to the machines.

*Contributions.* In this paper, we provide solution methods for the enumeration of the outcomes of pure Nash equilibria in a two-player perfect-recall game. We chose this specific class of extensive-form games, because it can be described with the most convenient notation and it can be compared to the only other result known in the literature. Our main objective is to introduce a new algorithmic approach that focuses exclusively on the outcomes of extensive-form games. We believe that this method can be extended to a larger class of games, but such analysis is left to future works.

Our new representation of extensive-form games describes a game as a graph of its outcomes. Based on this, a method is provided to determine whether an outcome is the realisation of a Nash equilibrium of a two-player extensive-form game. Since the utility of every outcome is known, it is possible to determine upper (resp., lower) bounds to the utility of Nash equilibria for any player by inspecting them once ordered from the best to the worst (resp., from worst to best), rather than enumerating them all. Moreover, it is possible to enumerate the Nash equilibria whose realisations fit any given set of constraints on the utility by simply checking during our enumeration procedure the outcomes that meet such constraints. The method is more efficient when the games have specific structures. Indeed, it performs very well in games where outcomes can be compared without explicitly constructing the sequences of actions that lead to them. This is the case of games with a regular structure as described by examples later in the manuscript.

*Summary.* The paper is structured as follows. Section 2 presents extensive-form games with perfect information with a generic number of players. In Sect. 2.2, we describe a known

method to bound the utility of the Nash equilibria of a two-player game. Section 3 introduces a new graph formulation of extensive-form games that allows a new characterisation of pure Nash equilibria in two-player games with perfect information. An equivalent formulation is discussed in Sect. 3.3 and used in Sect. 4 to achieve the following objectives: to identify a Nash equilibrium, to enumerate all Nash equilibria, to find an upper bound to their utility which is compared to the methods in literature (cf. [16]) and to identify the best and the worst Nash equilibrium for a player. Section 5 presents extended computational experiments of such methods on a given dataset of games. Section 6 ends the paper with some insights and possible research directions to extend the results to more generic extensive-form games. The reader can find examples of games in Appendix A to visualize the notions introduced in the article and detailed proofs of the theorems in Appendix B.

## 2 Extensive-Form Games

**Notation.** In this work we will use the following notation for vectors. A *vector*  $a = (a^1, \dots, a^n)$  is an ordered sequence of elements  $a^k$  with  $k \in \{1, \dots, n\}$ . Given two vectors  $a = (a^1, \dots, a^n)$  and  $b = (b^1, \dots, b^n)$ , a *concatenation* of vectors  $a + b = (a^1, \dots, a^n, b^1, \dots, b^n)$  is represented by the operator  $+$ . Given a concatenation of vectors  $a + b$ ,  $a$  is said to be a *prefix* of  $a + b$ , and that  $a \leq a + b$ . A vector with no elements is called the *empty vector*  $\emptyset$  and  $\emptyset \leq a$  for all  $a$ . Two vectors  $a$  and  $a'$  might have a common prefix. We denote by  $c = a \cap a'$  the *longest common prefix* of  $a$  and  $a'$ , i.e., the longest sequence such that  $c \leq a$  and  $c \leq a'$ .

### 2.1 Definitions

We consider discrete, finite extensive-form games with perfect information, whose feature is to have all information sets as singletons (cf. [1, 24]). We will describe this class of games with an operative definition which appears more suitable to our algorithmic framework than the standard one appearing in the literature. Note that, even if our definition is written specifically for finite extensive-form games with perfect information, it is indeed compliant with other ones appearing in the literature (cf. [25–27]). They are provided for a generic  $N$ -players extensive-form game, even though this manuscript tackles the 2-player case only.

Let  $\mathcal{I} = \{1, \dots, N\}$  be the set of players acting in a sequence of time instants. At each time instant, a player observes a history of actions  $h'$  and picks an action  $a \in \mathcal{A}(h')$ . In particular, at the beginning, the first player to act observes no actions  $h^0 = \emptyset$  and thus can act within a set of  $M$  actions  $\mathcal{A}(h^0) = \{a_1^0, \dots, a_M^0\}$ . Let us say that for instance she picks  $a^0 = a_m^0 \in \mathcal{A}(h^0)$ ; the second player observes  $h^1 = (a^0)$  and can thus pick an action  $a^1 \in \mathcal{A}(h^1)$ . Iterating at each instant this procedure, we define the set of histories  $H'$ . Accordingly, there is a function  $\mathcal{A} : h' \in H' \mapsto A$  that maps every history  $h'$  to the set of actions  $A$  available to the player observing history  $h'$ . The game ends when there is no actions left. Formally, there is a subset  $H \subset H'$  such that  $\mathcal{A}(h) = \emptyset$  for all  $h \in H$ . Such terminal histories are called *outcomes*. Every outcome  $h \in H$  is evaluated by a function  $u_i$  that maps it to the value  $u_i(h) \in \mathbb{R}$  assigned by player  $i$  to  $h$ .

**Definition 1** (extensive-form game) An extensive-form game is a tuple  $\Gamma = \langle \mathcal{I}, \mathcal{A}, H', H, P, u \rangle$ , where:

- $\mathcal{I} = \{1, \dots, N\}$  is the set of players;
- $H'$  is the set of histories with  $\emptyset \in H'$ ;

- $\mathcal{A} : h' \in H' \rightarrow A$  is a function that provides for every history a set of actions, i.e., for all  $a \in A = \mathcal{A}(h')$ , we have  $h' + (a) \in H'$ ;
- $H = \{h \in H' \mid \mathcal{A}(h) = \emptyset\} \subset H'$  is the set of outcomes;
- $P : H' \setminus H \rightarrow \mathcal{I}$  is a function that indicates which player  $P(h') \in \mathcal{I}$  acts after observing the history  $h' \in H' \setminus H$ ;
- $u = (u_i)_{i \in \mathcal{I}}$ , with  $u_i : H \rightarrow \mathbb{R}$ , is the utility function.

The size of a finite extensive form game is the number of outcomes. In the literature, the standard representation of the game is the tree of possible histories (cf. Appendix A.1). Hence, we alternatively call *node* a history observed by a player. The standard definition of a *strategy* identifies the single action chosen by a player after observing a history. In this paper, we consider pure strategies as we leave to future works the analysis and algorithmic solution of the games in mixed strategies.

**Definition 2** (strategy) Given a game  $\Gamma = \langle \mathcal{I}, \mathcal{A}, H', H, P, u \rangle$  and a player  $i \in \mathcal{I}$ , we pick all the histories at which the player acts:  $H_i = \{h' \in H' \setminus H \mid P(h') = i\}$ . A strategy  $s_i$  is a function  $s_i : h' \in H_i \mapsto a \in \mathcal{A}(h')$  that maps every observed history  $h' \in H_i$  to an action  $a \in \mathcal{A}(h')$  available to the player  $i$ . Let  $S_i$  denote the set of all strategies of player  $i$ .

If every player picks a strategy, we have a tuple of strategies  $s = \langle s_1, s_2, \dots, s_N \rangle$ , that we call *strategy profile*. If we consider a strategy profile, for every history there will be an action to be played. Eventually, this sequence of actions makes an outcome. Such outcome is defined as the *realisation* of the strategy profile. In the following sections we write  $s \mapsto h$  to identify the unique realisation  $h$  of the strategy profile  $s$ . Moreover, with some abuse of notation, we write  $u(s)$ , i.e., the utility of a strategy profile  $s$ , to indicate  $u(s) = u(h : s \mapsto h)$ .

The solution of a game is an *equilibrium*, i.e., a situation in which players do not want to change their strategies. The *Nash equilibrium* is a combination of strategies for which the players find it convenient not to deviate unilaterally. More specifically, if the other players do not change their strategies  $s_{-i} = (s_j)_{j \in \mathcal{I} \setminus \{i\}}$ , the player  $i$  has no interest in changing her own strategy  $s_i$  because she would not improve her utility. This concept of Nash equilibrium is hereafter defined.

**Definition 3** (Nash equilibrium) Given a game  $\Gamma = \langle \mathcal{I}, H, u \rangle$ , we say that a strategy profile  $(\bar{s}_i)_{i \in \mathcal{I}}$  is a Nash equilibrium if for every  $i \in \mathcal{I}$  and for all  $s_i \in S_i$ :

$$u_i(\bar{s}_i, \bar{s}_{-i}) \geq u_i(s_i, \bar{s}_{-i}).$$

An extensive-form game with perfect information always admits at least one Nash equilibrium in pure strategies (cf. [6]).

## 2.2 Bounds for Nash Equilibria in Two-Player Games

The most efficient methods to provide an upper bound to the utility of Nash equilibria have been introduced for two-player extensive-form games. We suppose that the games are not zero-sum, since such subcategory of games has already been fully studied (cf. [16]).

Given a two-player extensive-form game  $\Gamma = \langle \mathcal{I} = \{1, 2\}, \mathcal{A}, H', P, u \rangle$ , we consider the following optimization problem [ST]:

$$\begin{aligned}
 [ST] : \quad & \bar{s}_1 \in \arg \max_{s_1 \in S_1} u_1(s_1, \bar{s}_2) \\
 & s.t. \quad \bar{s}_2 \in \arg \max_{s_2 \in S_2} u_2(s_1, s_2).
 \end{aligned}$$

This bilevel optimization problem has linear complexity in the number of the strategies of the game. We note however that they might still be exponential in the game size, i.e., in number of outcomes (cf. [16]). Indeed, let us for instance consider a game represented by a complete binary tree, i.e., a game in which every node has two actions. Given  $|H' \setminus H|$  the number of non-leaf nodes of the game tree, it is easy to show that there are  $2^{|H' \setminus H|}$  strategy profiles and  $|H' \setminus H| + 1$  outcomes. However, a different formulation of [ST] introduced in (cf. [16]) can be written in the form of a bi-level linear program with a number of variables and a number of constraint inequalities that grow linearly with respect to the tree size. Such formulation, which considers mixed strategies, is discussed in details in Sect. 4.2 and it is adapted to solve games in pure strategies.

An optimal solution of [ST] provides an upper bound to the utility of the first player of all the Nash equilibria of the game (cf. [28]).

**Theorem 1** *Let us consider  $\bar{S}$  the set of feasible solutions  $\bar{s} = (\bar{s}_1, \bar{s}_2) \in S_1 \times S_2$  of [ST] and the optimum value  $\bar{U}_1 = \sup_{\bar{s} \in \bar{S}} u_1(\bar{s})$ . Given a Nash equilibrium  $s^* \in S_1 \times S_2$  of the game  $\Gamma$ , it holds:*

$$\bar{U}_1 \geq u_1(s^*).$$

**Proof** It is sufficient to prove that any Nash equilibrium is a feasible solution of [ST]. Indeed, since  $s^*$  is a Nash Equilibrium, we have that for all  $s_2 \in S_2$ :

$$u_2(s_1^*, s_2^*) \geq u_2(s_1^*, s_2).$$

Therefore it holds that  $s_2^* = \arg \max_{s_2 \in S_2} u_2(s_1^*, s_2)$ . □

### 3 Graph Form

#### 3.1 A Representation of Extensive-Form Games as Graphs of Outcomes

In this section, we introduce a new representation of a two-player game in extensive form with perfect information as an undirected graph of its outcomes. The goal of this representation is to focus on the outcomes, in order to identify which of them are realisations of the Nash equilibria. We next put in relation the strategies of the players with the outcomes of the game.

If a player chooses a strategy, she limits the number of outcomes that are reachable by the other player. We formalise this observation by associating a strategy to a subset of outcomes that represent it. We recall that every strategy profile  $(s_1, s_2) \in S_1 \times S_2$  can be mapped to its realisation  $h : (s_1, s_2) \mapsto h$ . Given a strategy  $s_1 \in S_1$ , we consider the set of possible outcomes  $H(s_1)$  which are realisations of strategy profiles  $(s_1, s_2)$  which include  $s_1 \in S_1$  as a strategy of the first player and any  $s_2 \in S_2$  as a strategy of the second player.

**Definition 4** (outcomes of a strategy) Given a two-player game  $\Gamma = \langle \mathcal{I} = \{1, 2\}, H, u \rangle$  and a strategy  $s_1 \in S_1$ , the set of outcomes  $H(s_1) \subset H$  of strategy  $s_1$  is:

$$H(s_1) = \{h \in H \mid \exists s_2 \in S_2 : (s_1, s_2) \mapsto h\}.$$

In order to understand which elements belong to the set of outcomes of a strategy, we introduce a new property, called *compatibility*. This property enables to identify two outcomes that can be obtained by the same strategy chosen by a given player.

**Definition 5** (compatibility) Given a two-player game  $\Gamma = \langle \mathcal{I} = \{1, 2\}, H, u \rangle$ , we say that two outcomes  $h, h' \in H$  are compatible for player  $i \in \mathcal{I}$  if there is a strategy  $s_i \in S_i$  such that  $h \in H(s_i)$  and  $h' \in H(s_i)$ .

**Remark** Since we discuss only two-player games, i.e.,  $\mathcal{I} = \{1, 2\}$ , we arbitrarily choose one player to be the *first player* (e.g.,  $i_1 = 1$ ) and one to be the *second player* (e.g.,  $i_2 = 2$ ). Later in this section it is shown that such choice can be arbitrary. If not specified, we refer to two outcomes as *compatible* if they are compatible for player 1. If two outcomes  $h, h' \in H$  are compatible, there are a strategy  $s_1 \in S_1$  and two strategies  $s_2, s'_2 \in S_2$  such that  $(s_1, s_2) \mapsto h$  and  $(s_1, s'_2) \mapsto h'$ .

If two outcomes can be produced by the same strategy, the first player always takes the same decisions at every node. Lemma 2 proves that this condition is not only necessary but sufficient. Formally, given two outcomes  $h, h' \in H$  it is necessary to observe at which node the history starts to be different; such node is identified by their longest common prefix  $h \cap h'$ .

**Lemma 2** *We consider a two-player game  $\Gamma = \langle \mathcal{I} = \{1, 2\}, H, P, u \rangle$ . Two outcomes  $h, h' \in H$  are compatible if and only if  $P(h \cap h') = 2$ .*

**Proof** (i) First we prove that  $P(h \cap h') = 2$  implies that  $h$  and  $h'$  are compatible, then (ii) we prove that  $P(h \cap h') = 1$  implies that  $h$  and  $h'$  are not compatible.

(i) Let us suppose that  $P(h \cap h') = 2$ . We need to define  $s_1 \in S_1$  and  $s_2, s'_2 \in S_2$  such that  $(s_1, s_2) \mapsto h$  and  $(s_1, s'_2) \mapsto h'$ . We recall that a strategy of the first player is a function associating an action to each partial history  $h'' \in H'$  in the tree observed by the first player, i.e., such that  $P(h'') = 1$ . For each  $h'' \leq h$  ( $h'' \leq h'$ ) if  $P(h'') = 1$  we define a strategy  $s_1 \in S_1$  such that  $h'' + (s_1(h'')) \leq h$  ( $h'' + (s_1(h'')) \leq h'$ ). Analogously, we define a strategy  $s_2 \in S_2$ , such that for each  $h'' \leq h$  with  $P(h'') = 2$  we have  $h'' + (s_2(h'')) \leq h$ , and a strategy  $s'_2 \in S_2$ , such that for each  $h'' \leq h'$  with  $P(h'') = 2$  we have  $h'' + (s'_2(h'')) \leq h'$ . For any other  $h'' \not\leq h$  and  $h'' \not\leq h'$  we take an arbitrary decision for defining  $s_1, s_2, s'_2$ . By construction, we have  $(s_1, s_2) \mapsto h$  and  $(s_1, s'_2) \mapsto h'$ .

(ii) Let us now suppose that  $P(h \cap h') = 1$ . Let  $(s_1, s_2)$  and  $(s'_1, s'_2)$  be two strategy profiles such that  $(s_1, s_2) \mapsto h$  and  $(s'_1, s'_2) \mapsto h'$ . It is impossible to have  $s_1 = s'_1$  since  $P(h \cap h') = 1$  implies  $s_1(h \cap h') \neq s'_1(h \cap h')$ , from which we can conclude that  $h$  and  $h'$  are not compatible. □

Based on the definition of compatibility, it is possible to build a graph of compatibilities among all the outcomes of a game  $\Gamma$ , or the graph form for short.

**Definition 6** (graph form) The graph of compatibility of a two-player game  $\Gamma = \langle \mathcal{I} = \{1, 2\}, H, u \rangle$  is a tuple  $\Gamma = \langle H, E, u \rangle$ , where  $H$  is the set of outcomes as nodes of the graph,  $E \subset H^2$  is the set of edges connecting any two compatible outcomes and  $u : H \rightarrow \mathbb{R}^2$  is the utility function that assigns a pair of weights to every node.

**Remark** In the following of the article we sometimes omit the transformation of an extensive-form game  $\Gamma$  into its graph form  $\langle H, E, u \rangle$ , which can be performed by application of Lemma 2, and therefore we directly introduce the game by representing it in its graph form  $\Gamma = \langle H, E, u \rangle$ .

### 3.2 Characterisation of Nash Equilibria in the Graph Form of the Game

Given a game  $\Gamma$ , let us characterise the outcomes  $H(s_1)$  of a strategy  $s_1 \in S_1$  on its graph form. By definition, such outcomes are all compatible with one another. Let us consider the nodes on the graph corresponding to the outcomes  $H(s_1)$ ; they induce a clique and, as we show next, such clique is maximal.



**Lemma 3** Consider a two-player game  $\Gamma = \langle \mathcal{I} = \{1, 2\}, H, u \rangle$  with its graph form  $\Gamma = \langle H, E, u \rangle$ . For every strategy  $s_1 \in S_1$ , the set  $H(s_1) \subset H$  forms a maximal clique of the graph  $\langle H, E, u \rangle$ .

**Proof** According to Definition 5, we have that  $H(s_1)$  induces a clique on the graph. Consider any outcome  $h \in H \setminus H(s_1)$ . Since  $h \notin H(s_1)$  there is a partial history  $h'' \in H'$ , with  $h'' \leq h$  and  $P(h'') = 1$ , such that the subsequent action  $a_{k+1} \in \mathcal{A}(h'')$  is not chosen by strategy  $s_1$ , i.e.,  $a_{k+1} \neq s_1(h'')$ . Consider now an outcome  $h' \in H(s_1)$  such that  $h'' + (s_1(h'')) \leq h'$ .

Since  $P(h \cap h') = 1$ , from Lemma 2  $h$  and  $h'$  are not compatible. Since this argument holds for every  $h \in H \setminus H(s_1)$ , we have that  $H(s_1)$  forms a maximal clique.  $\square$

We observe that to every maximal clique of the graph there is at least one strategy whose set of outcomes corresponds to it. We prove that it is always true in Lemma 4.

**Lemma 4** Let us consider a two-player game  $\Gamma = \langle \mathcal{I} = \{1, 2\}, H, u \rangle$  with its graph form  $\Gamma = \langle H, E, u \rangle$ . For every set of vertices  $\mathcal{C}$  that induces a maximal clique on the graph  $\langle H, E \rangle$ , there is a strategy  $s_1 \in S_1$  such that  $\mathcal{C} = H(s_1)$ .

**Proof** We first show that  $\mathcal{C} \subset H(s_1)$  for a given  $s_1 \in S_1$ . For this, we consider a strategy  $s_1 \in S_1$  such that, for all  $h \in \mathcal{C}$  and all  $h'' \leq h$  such that  $P(h'') = 1$ , we have that  $h'' + s_1(h'') \leq h$ . Such strategy exists and it is defined by applying a procedure similar to the one used in the proof of Lemma 2; we recall that  $P(h \cap h') = 2$  for each pair of compatible outcomes  $h, h' \in \mathcal{C}$ .

Therefore  $\mathcal{C} \subseteq H(s_1)$ . But, from Lemma 3,  $H(s_1)$  induces a maximal clique and thus  $\mathcal{C} = H(s_1)$ .  $\square$

Lemmas 3 and 4 establish that there exists a bijection between a partition of the set of strategies of the game and the set of maximal cliques in the graph form. An illustration is given in Fig. 7b. A similar result is obtained for the set of strategies of the second player on the complementary graph.

**Lemma 5** For every two-player game  $\Gamma = \langle \mathcal{I} = \{1, 2\}, H, P, u \rangle$  with its graph form  $\Gamma = \langle H, E, u \rangle$ , there is a bijection between a partition of the set of strategies of the second player  $S_2$  and the set of maximal cliques of the complementary graph  $\langle H, E^C \rangle$ , where  $E^C = \{(h, h') \in H^2 \mid h \neq h', (h, h') \notin E\}$ .

**Proof** Given two outcomes  $h, h' \in H$ , we have that  $P(h \cap h') \in \{1, 2\}$ . Therefore  $P(h \cap h') = 2$  if and only if  $P(h \cap h') \neq 1$ . Lemma 2 thus determines that the graph form of the second player is complementary to the one of the first player. The result follows from Lemmas 3 and 4.  $\square$

Choosing a strategy for the first player implies picking a maximal clique in the graph  $\langle H, E \rangle$ . Furthermore, one can observe that with an analogous method it is possible to build a graph for the strategies of the second player. However, thanks to Lemma 5 it is not necessary to perform further computations, because such graph is complementary to  $\langle H, E \rangle$ .

*Enabling Strategy.* We observe that the minimal representation of strategies as sets of outcomes can be obtained by applying the concept of *enabling strategy* of [19] to games with perfect information. Given a player  $i \in \mathcal{I}$  and an outcome  $h \in H$ , an *enabling strategy*  $S_i(h) = \{s_i \in S_i \mid h \in S_i\}$  is the set of strategies of player  $i \in \mathcal{I}$  leading to outcome  $h \in H$ . Enabling strategies are the minimal representation of the set of strategies with respect to specific properties of the space of strategies. In games with perfect information and two



outcomes  $h_1, h_2 \in H$ , there is only one player  $i \in \mathcal{I}$  such that  $S_i(h_1) \neq S_i(h_2)$ . The previous Lemmas can be obtained by combining this property with the known properties of enabling strategies (cf. [19]).

As anticipated in Sect. 2, we are interested in identifying Nash equilibria outcomes in pure strategies. We recall that a Nash equilibrium is a strategy profile in which none of the players is interested in changing her own strategy unilaterally.

Best responses and outcomes. The standard way to define a best response for an extensive-form game entails to refer to its equivalent strategic form. In this case, if a player picks a strategy, the other player will choose a *best response*, i.e., a strategy such that her utility is maximized. A Nash equilibrium can be identified as a mutual best response accordingly. We now provide a connection between this fundamental definition of equilibrium based on the strategic form and properties of the outcomes in the graph form of the game. In fact, every designated outcome  $h$  corresponds to one or more strategies  $s_1$  for the first player identified by a maximal clique  $C_1$  on the graph form which includes  $h$ . In turn, for every element  $h' \in C_1$ , the second player can choose a strategy  $s_2$  whose corresponding set of outcomes  $C_2$  on the complementary graph includes  $h' \in C_2$ . The second player has an incentive to pick the element within  $C_1$  that maximises her utility, a condition that thus  $h$  must fulfill in order to avoid deviations from the second player. A similar argument can be put forward by inverting the players. Note that a single outcome may correspond to multiple strategy pairs. Thus, in the graph form, determining if an outcome corresponds to a Nash equilibrium means to answer to the question if there exists at least a pair of maximal cliques respectively on the graph and on the complementary graph containing this outcome, and such that their intersection is a mutual best response in the corresponding strategic form.

The best response of the second player leads to the outcome which is most preferred by the second player within the maximal clique  $C_1$  chosen by the first player. If a vertex  $h \in H$  corresponds to the outcome of a best response of the second player, two conditions must hold. First, there must be a maximal clique  $C_1$  which includes it, i.e.,  $h \in C_1$ , so that the second player can choose it. Second, such maximal clique  $C_1$  should not include all the nodes  $X^h$  which are preferred to  $h$  by the second player, so that the second player has an incentive to choose  $h$  and no other outcome. Whether such clique exists is the problem [MC] formalised hereafter.

**Problem 6 (MC)** *Existence of a maximal clique including  $h$  and excluding  $X^h$ .*

*INSTANCE:*  $\langle H, E, h, X^h \rangle$  defining a graph  $\langle H, E \rangle$ , a vertex  $h \in H$  and a subset of vertices  $X^h \subset H$  with  $h \notin X^h$ .

*QUESTION:* Is there a vertex set  $C \subset H \setminus X^h$  with  $h \in C$  that induces a maximal clique on  $\langle H, E \rangle$ ?

We would like to know if an outcome  $h$  is the realisation of a Nash equilibrium. Hence, a maximal clique including the corresponding vertex  $h$  and excluding  $X^h = \{h' \in H | u_2(h') > u_2(h)\}$  ensures that the first player has a strategy to induce the second player onto the desired outcome  $h$ .

**Example** In the game of Fig. 6 the second player prefers  $X^h = \{h_2, h_3, h_4\}$  to  $h = h_6$ . In the graph of Fig. 7a we observe that there is no maximal clique including  $h_6$  and excluding all the elements of  $X^h$ . Therefore  $h_6$  cannot correspond to a best response of the second player.

Furthermore, by applying the same arguments on the complementary graph we conclude that it is possible to determine whether an outcome is the realisation of a best response also of the first player. Specifically, it is necessary to identify a maximal clique on the complementary

graph such that the vertices corresponding to the outcomes preferred by the first player are excluded.

**Example** In the graph of Fig. 7a the first player prefers  $X^h = \{h_2, h_3, h_5, h_6\}$  to  $h = h_1$ . There is no maximal clique on the complementary graph of Fig. 7a, i.e., there is no independent set, that includes  $h_1$  and none of the elements in  $X_h$ . Therefore  $h_1$  cannot be a best response of the first player.

Theorem 7 combines these findings and provides a characterisation of a Nash equilibrium in the graph form.

**Theorem 7** *Let us consider a two-player game in its graph form  $\Gamma = \langle H, E, u \rangle$  and an outcome  $h \in H$ . We consider  $X_1^h = \{h' \in H \mid u_1(h') > u_1(h)\}$  and  $X_2^h = \{h' \in H \mid u_2(h') > u_2(h)\}$ , the sets of outcomes preferred to  $h$ , respectively, by the first and the second player. The outcome  $h \in H$  is a realisation of a Nash equilibrium if and only if the problem [MC] has true as answer both providing as input  $\langle H, E, h, X_2^h \rangle$  and  $\langle H, E^C, h, X_1^h \rangle$ .*

The proof is provided in Appendix B.1. Such result allows us to develop methods that compute Nash equilibria without listing all the strategies of the players, which are often in exponential number with respect to the size of the game (cf. Section 2). Such methods will be discussed in the following sections.

### 3.3 Analysis of the Main Problem and its Complexity

The complexity of identifying a Nash equilibrium outcome on a game in its graph form depends on the complexity of solving two instances of problem [MC]: one with input  $\langle H, E^C, h, X_1^h \rangle$  and another with input  $\langle H, E, h, X_2^h \rangle$ , as defined in the previous section. In this section, we evaluate the complexity of problem [MC] with input  $\langle H, E, h, X^h \rangle$  for a generic graph  $\langle H, E \rangle$ . More specifically, we first define a variant [EC] of problem [MC] and then we show that the two problems are equivalent, i.e., that a solution of problem [EC] provides a solution of problem [MC] and conversely.

Let us consider a generic problem [MC] with input  $\langle H, E, h, X^h \rangle$ . First, we argue that when solving [MC] the problem can be restricted to the neighbourhood of  $h \in H$ ,  $V^h = \{h', (h, h') \in E\}$ . Indeed, let us suppose that there is a maximal clique induced by a vertex set  $C \subset H$  with  $h \in C$  excluding  $X^h \subset H$ . Since the clique is maximal, it must hold that, for every vertex  $h' \in X^h$ , there is at least one vertex  $h \in C$  which is not connected to  $h'$ , i.e., such that  $(h, h') \notin E$ . Those vertices in  $X^h$  who are not in the neighborhood  $V^h$  always fulfill this property, as  $h \in C$ . Therefore, instead of considering all the vertices in  $X^h$ , we can restrict the problem to  $X = X^h \cap V^h$ . With this argument, we conclude that the vertex set  $C$  belongs to the neighborhood  $V^h$ .

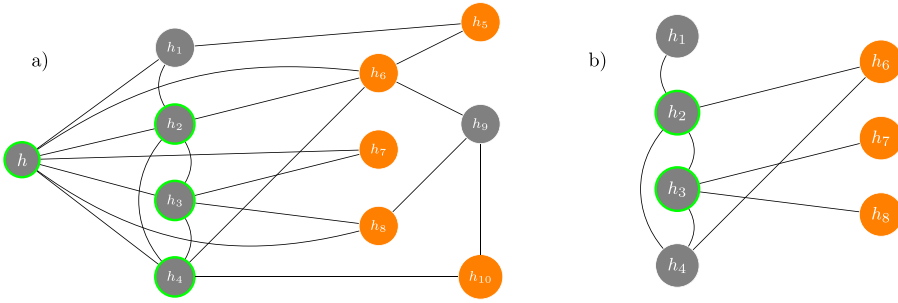
**Example** Consider the graph of Fig. 1a, in which the set  $C = \{h, h_2, h_3, h_4\}$  induces a maximal clique. The vertices  $X^h \setminus V^h = \{h_5, h_{10}\}$  are not connected to  $h$  and the vertex  $H \setminus X^h \setminus V^h = \{h_9\}$  can never belong to the vertex set  $C$  that induces the maximal clique. Therefore we can restrict the problem from  $H$  to  $\{h_1, h_2, h_3, h_4, h_6, h_7, h_8\}$ .

Let us thus consider a slightly different problem [EC].

**Problem 8 (EC)** *Existence of an excluding clique.*

*INSTANCE:*  $\langle V, X, E \rangle$  defining a graph  $\langle V \cup X, E \rangle$  with  $V \cap X = \emptyset$ .

*QUESTION:* Is there a vertex set  $C \subset V$  that induces a clique on  $\langle V, E \rangle$  that is maximal in  $\langle C \cup X, E \rangle$ ?



**Fig. 1** Equivalence of problems  $[MC]$  and  $[EC]$ . **a** Let us consider problem  $[MC]$  with  $H = \{h_1, h_2, h_3, h_4, h_5, h_6, h_7, h_8, h_9, h_{10}\}$  and  $X^h = \{h_5, h_6, h_7, h_8, h_{10}\}$ . A maximal clique that solves  $[MC]$  is  $C = \{h, h_2, h_3, h_4\}$ ; **b** Let us consider problem  $[EC]$  with  $V = \{h_1, h_2, h_3, h_4\}$  and  $X = \{h_6, h_7, h_8\}$ . A clique that solves  $[EC]$  is  $C' = \{h_2, h_3\}$

Theorem 9 shows that the problem  $[MC]$  with input  $\langle H, E, h, X^h \rangle$  can be solved by means of problem  $[EC]$  with a different input derived by the input of problem  $[MC]$ .

**Example** Indeed let us consider the problem  $[MC]$  depicted in Fig. 1a and its restriction to the neighborhood of  $h$  of Fig. 1b. The problem  $[MC]$  requires to identify a maximal clique that has no elements in  $X^h$  and includes  $h$ , such as  $\{h, h_2, h_3, h_4\}$  or  $\{h, h_1, h_2\}$ . The problem  $[EC]$  requires to identify a vertex set  $C'$  on  $V = \{h_1, h_2, h_3, h_4\}$  that induces a clique such that for all elements in  $X = \{h_6, h_7, h_8\}$  there is at least one element in  $C'$  not connected to it. Such cliques are  $\{h_1\}$ ,  $\{h_1, h_2\}$ ,  $\{h_2, h_3\}$ ,  $\{h_3, h_4\}$  and  $\{h_2, h_3, h_4\}$ . For instance, let us consider  $\{h_2, h_3\}$ : the vertex  $h_6$  is not connected to  $h_3$ , while the vertices  $h_7$  and  $h_8$  are not connected to  $h_2$ .

**Proposition 9** Let us consider a graph  $\langle H, E \rangle$ , a subset of vertices  $X^h \subset H$  and a vertex  $h \in H \setminus X^h$ . Let us define  $V^h = \{h' | (h, h') \in E\}$ ,  $X = X^h \cap V^h$ ,  $V = V^h \setminus X$  and  $E|_{V \cup X} = \{(h', h'') \in E | h' \in V, h'' \in V \cup X\}$ . The problem  $[MC]$  with input  $\langle H, E, h, X^h \rangle$  has true as answer if and only if the problem  $[EC]$  with input  $\langle V, X, E|_{V \cup X} \rangle$  has.

The proof is provided in Appendix B.2.

We observe that problem  $[EC]$  requires to prove the existence of a clique rather than identifying a maximal clique, as in problem  $[MC]$ . Moreover, the input of  $[EC]$  is a smaller graph induced by the neighborhood of one vertex,  $h$ , in graph of  $[MC]$ . Therefore, from now on, let us focus on the analysis of  $[EC]$ . In the following theorem, we prove that  $[EC]$  is NP-complete. Indeed, we reduce it to the problem of the existence of the dominating clique  $[DC]$ , which is known to be NP-complete (cf. [29]).

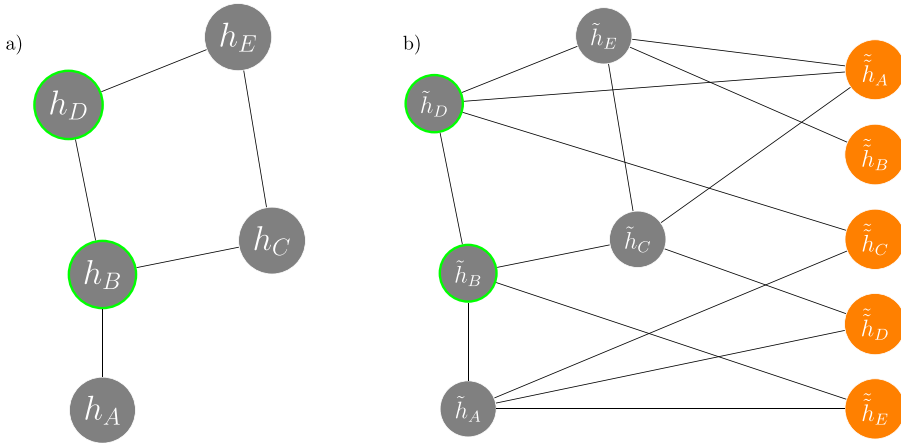
**Problem 10 (DC) Existence of a dominating clique**

*INSTANCE:* A graph  $\langle H, E \rangle$ .

*QUESTION:* Is there a vertex set  $C \subset H$  that induces a clique on the graph such that for every vertex  $h'' \in H \setminus C$  there is a vertex  $h' \in C$  such that  $(h', h'') \in E$ ?

**Theorem 11** In a generic graph the problem  $[EC]$  is NP-complete.

**Proof** We present next a polynomial reduction from  $[DC]$  to  $[EC]$ . The argument of the proof is illustrated in Fig. 2. We consider the problem  $[DC]$  with input  $\langle H, E \rangle$  and define two vertex sets  $V = \tilde{H}$  and  $X = \tilde{\tilde{H}}$ , where  $\tilde{H}$  and  $\tilde{\tilde{H}}$  are copies of set  $H$ . Let us also define



**Fig. 2** Reduction. **a** Problem [DC] with  $H = \{h_A, h_B, h_C, h_D, h_E\}$  and solution  $C = \{h_B, h_D\}$ ; **b** Problem [CL] with  $V = \tilde{H}$  and  $X = \tilde{H}$ ; the solution is given by  $C = \{\tilde{h}_B, \tilde{h}_D\} \subset V$

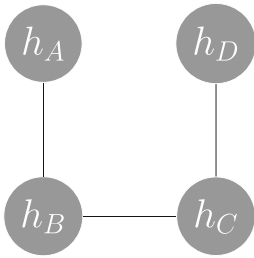
a set of edges  $E' = \{(i, j) | i, j \in V, (i, j) \in E\} \cup \{(i, j) | i \in V, j \in X, (i, j) \notin E\}$ . We consider the problem [EC] with input  $\langle V, X, E' \rangle$ . By construction, the input has size  $O(|H|^2)$ , i.e., it is polynomial in the size of the input of problem [DC]. We prove now that [DC] admits answer true if and only if the same happens for [EC] with input  $\langle V, X, E' \rangle$ . If [DC] has answer true, there exists a vertex set  $C \subset H$  that: (a) induces a clique on  $\langle H, E \rangle$ ; (b) and such that, for each  $j \in H \setminus C$ , there is a  $i \in C$  such that  $(i, j) \in E$ . From (a) and the definition of sets  $V$  and  $E'$ , there is a copy of  $C \subset V$  defining a clique in graph  $\langle V \cup X, E' \rangle$ . Also, from the definition of  $E'$  and from (b), for each  $j \in X$  there is a  $i \in C$  such that  $(i, j) \notin E'$ . Therefore  $C$  provides also an answer true for [EC].

We now prove that an answer true for [EC] provides an answer true also to [DC]. A vertex set  $C \subset V$  that induces a clique on  $\langle V \cup X, E' \rangle$ , clearly defines a clique on  $\langle H, E \rangle$ . It holds that for all  $j \in X$  there is  $i \in C$  such that  $(i, j) \notin E'$ . Since  $X = \tilde{H}$  and  $V = \tilde{H}$ , with  $\tilde{H}$  and  $\tilde{H}$  copies of  $H$ , we have that for all  $j \in H \setminus C$  there is a vertex  $i \in C$  such that  $(i, j) \in E$ . Therefore  $C$  provides also a solution true for [DC].  $\square$

*On the Graph of an Extensive-Form Game.* We have just proved that problem [EC] is NP-complete for a generic graph. However, the graph generated by an extensive-form game is not a generic one. Indeed, it is possible to identify a graph that corresponds to no extensive-form games. The graph  $\langle H, E \rangle$  of Fig. 3 does not represent any extensive-form game (cf. Appendix A.5).

### 4 New Methods for the Identification of Nash Equilibria

In this section, the theoretical results introduced in Sect. 3 are applied to derive new methods for the computation of outcomes of Nash equilibria in two-player extensive-form games with perfect information. Theorem 7 provides a necessary and sufficient condition for an outcome to be a realisation of a Nash equilibrium. This result is the pillar for the development of computational methods to the following questions: (i) whether it is possible to enumerate



$h$	$\langle H, E \rangle$		$\langle H, E^C \rangle$	
	V	X	V	X
$h_A$	$\emptyset$	$\{h_B\}$	$\{h_C, h_D\}$	$\emptyset$
$h_B$	$\{h_A, h_C\}$	$\emptyset$	$\emptyset$	$\{h_D\}$
$h_C$	$\{h_D\}$	$\{h_B\}$	$\emptyset$	$\{h_A\}$
$h_D$	$\emptyset$	$\{h_C\}$	$\{h_B\}$	$\{h_A\}$

**Fig. 3** Counterexample. Preferences of the players over the outcomes are respectively  $u_1 : h_A \succ_1 h_C \succ_1 h_D \succ_1 h_B$  and  $u_2 : h_B \succ_2 h_C \succ_2 h_A \succ_2 h_D$

the Nash equilibria and (ii) whether the realisation of Nash equilibria can achieve a value of utility with a given range of values.

In what follows, we exploit the results of Sect. 3.3 and introduce a linear system that enables to determine if an outcome is the realisation of a Nash equilibrium. Relying on this linear system, we introduce a method to enumerate all Nash equilibria outcomes in Sect. 4.1, a method to provide an upper bound to their utility in Sect. 4.2 and a method to provide the best or worst Nash equilibrium outcomes for any player in Sect. 4.3. The latter is compared to the one provided by the optimization problem by [16].

### 4.1 Nash Equilibrium

The condition to verify if an outcome is the realisation of a Nash equilibrium introduced in Theorem 7 requires to solve two instances of problem [EC] with input  $\langle V \cup X, E \rangle$  such that  $V \cap X = \emptyset$ . We recall that the solution obtained by solving the problem allows to identify some possible outcomes  $\mathcal{C} \subset V$  of a strategy within the set  $V$  such that we have the guarantee that the elements  $X$  preferred by the opponent are not included. We provide a formulation [CL] of problem [EC]:

$$[CL] : \quad x_i + x_{i'} \leq 1, \quad \forall i, i' \in V, (i, i') \notin E, \quad (CL-1)$$

$$\sum_{i \in V | (i, j) \notin E} x_i \geq 1, \quad \forall j \in X, \quad (CL-2)$$

$$x_i \in \{0, 1\}, \quad \forall i \in V. \quad (CL-3)$$

Formulation [CL] models any feasible solution of problem [EC]:  $x_i = 1$  if and only if  $i \in \mathcal{C}$ . Constraints (CL – 1) impose that  $\mathcal{C}$  induces a clique, while constraints (CL – 2) guarantee that every vertex  $j \in X$  is not connected to at least one vertex  $i \in \mathcal{C}$ . Any solution to the linear system [CL] provides a solution to problem [EC] with input  $\langle V \cup X, E \rangle$ .

Theorem 7 imposes to solve two instances of the problem [MC] to determine if an outcome is the realisation of a Nash equilibrium. We thus exploit the fact that the problem [MC] can be modeled by formulation [EC] and define a unique linear system [NE] that allows to determine if an outcome is the realisation of a Nash equilibrium. Let us consider a two-player game in its graph form  $\Gamma = \langle H, E, u \rangle$  and an outcome  $h \in H$ . Let us define the following sets:

- $X_1 = \{h' \in H \mid (h', h) \notin E, u_1(h') > u_1(h)\}$ , the set of outcomes compatible with  $h$  in the complementary graph, that are preferred by the first player to  $h$ ;
- $X_2 = \{h'' \in H \mid (h'', h) \in E, u_2(h'') > u_2(h)\}$ , the set of outcomes compatible with  $h$  in the graph, that are preferred by the second player to  $h$ ;
- $V_1 = \{h' \in H \setminus X_2 \mid (h', h) \in E\}$ , the set of outcomes compatible with  $h$  in the graph, such that  $h$  is preferred to them for the second player;
- $V_2 = \{h'' \in H \setminus X_1 \mid (h'', h) \notin E\}$ , the set of outcomes compatible with  $h$  in the complementary graph, such that  $h$  is preferred to them for the first player.

The outcome  $h \in H$  is a realisation of a Nash equilibrium if and only if the system  $[NE]$  provides a solution:

$$\begin{aligned}
 [NE] : \quad & x_i + x_{i'} \leq 1, \quad \forall i, i' \in V_1, (i, i') \notin E, \\
 & x_i + x_{i'} \leq 1, \quad \forall i, i' \in V_2, (i, i') \in E, \\
 & \sum_{i \in V_1, (i, j) \notin E} x_i \geq 1, \quad \forall j \in X_2, \\
 & \sum_{i \in V_2, (i, j) \in E} x_i \geq 1, \quad \forall j \in X_1, \\
 & x_i \in \{0, 1\}, \quad \forall i \in V_1 \cup V_2.
 \end{aligned}$$

Given a game  $\Gamma = \langle H, E, u \rangle$ , by applying  $[NE]$  to every  $h \in H$  it is possible to enumerate all the realisations of the Nash equilibria. In the following, we propose an enumeration algorithm  $[EA]$  that iterates over all the outcomes and then solves  $[NE]$  for every outcome.

---

#### Algorithm 1 [EA] Enumeration algorithm

---

```

INPUT: Game in graph form  $\Gamma = \langle H, E, u \rangle$ 
 $NE \leftarrow \emptyset$ 
for  $h \in H$  do
     $X_1 = \{h' \in H \mid (h', h) \notin E, u_1(h') > u_1(h)\}$ 
     $X_2 = \{h'' \in H \mid (h'', h) \in E, u_2(h'') > u_2(h)\}$ 
     $V_1 = \{h' \in H \setminus X_2 \mid (h', h) \in E\}$ 
     $V_2 = \{h'' \in H \setminus X_1 \mid (h'', h) \notin E\}$ 
    solve  $[NE]$  giving  $\langle V_1, V_2, X_1, X_2, E \rangle$  as input
    if system  $[NE]$  has a feasible solution then
         $NE \leftarrow NE \cup \{h\}$ 
    end if
end for
OUTPUT:  $NE$ 
    
```

▷ for every outcome of the game  
 ▷ vertex sets and excluding sets  
 ▷ for player 1 and player 2  
 ▷ test if  $h$  is an outcome of a Nash equilibrium  
 ▷ Update the set of Nash equilibria outcomes

---

## 4.2 Upper Bounds for Nash Equilibria

In this section, we compare two methods to compute upper bounds for the utility of the first player when a Nash equilibrium is played. The most known formulation for extensive-form games is the one introduced by [16], which provides a Nash equilibrium for zero-sum games. Recently, this formulation has been proven to provide an upper bound to the utility of the first player of any Nash equilibrium (cf. [28]). This method is based on the concept of *sequence*, which is a vector of actions played by a same player. Given a game  $\Gamma = \langle \mathcal{I}, \mathcal{A}, H', P, u \rangle$  and a history  $h' \in H'$ , we denote by  $seq_i = (h_{ik})$  a sequence of actions played by player

$i$  according to  $h'$ . We write  $h' = (seq_1, seq_2)$  to show that to every history  $h'$  correspond two sequences  $seq_1$  and  $seq_2$ . We consider  $\Lambda_1$  and  $\Lambda_2$ , respectively, the set of all sequences of the first and second player. Let  $x \in \{0, 1\}^{|\Lambda_1|}$  and  $y \in \{0, 1\}^{|\Lambda_2|}$  be the vectors defining the probability for a sequence to be played. We define the matrix  $U^i : \Lambda_1 \times \Lambda_2 \rightarrow \mathbb{R}$  that maps each couple of sequences to the utility of player  $i$ :  $U^i_{seq_1, seq_2} = u_i(h)$  for all  $h = (seq_1, seq_2) \in H$ ;  $U^i_{seq_1, seq_2} = 0$  if  $h' = (seq_1, seq_2) \notin H$ . The utilities of the players can thus be written in the form  $x^T U^1 y$  and  $x^T U^2 y$ . The formulation defining the set of possible sequences is constrained by the fact that if an action is taken at a node of the game, such decision must be considered also in the following ones.

All such causal constraints, written  $Ex = e$  and  $Fy = f$ , respectively, for the first and the second player, will be built according to the same principle. Finally, the upper bound of any Nash equilibrium is given by the solution of the following bilevel problem denoted by [VS].

$$\begin{aligned}
 [VS] : \quad & u_1^{VS} = \max_x \quad x^T U^1 \bar{y} \\
 & \text{s.t. } Ex = e, \\
 & \quad x \in [0, 1]^{|\Lambda_1|}, \\
 & \quad \bar{y} = \arg \max_y \quad x^T U^2 y \\
 & \quad \text{s.t. } \quad Fy = f, \\
 & \quad \quad y \in [0, 1]^{|\Lambda_2|}.
 \end{aligned}$$

The optimization problem [VS] has size  $O(|H|)$  (cf. [16]). We now formulate [VS] as a linear optimization problem. Note that, as anticipated in Sect. 2, we are only interested in solutions corresponding to pure strategies. Therefore we can add the integral constraints:

$$x_j \in \{0, 1\} \quad \forall j \in \Lambda_1, \tag{1}$$

$$y_k \in \{0, 1\} \quad \forall k \in \Lambda_2. \tag{2}$$

We will also introduce the variable  $w_{jk} \in [0, 1]$ , which allows to linearise the formulation by rewriting  $x_j \cdot y_k = w_{jk}$  as:

$$x_j \geq w_{jk} \quad \forall j \in \Lambda_1, k \in \Lambda_2, \tag{3}$$

$$y_k \geq w_{jk} \quad \forall j \in \Lambda_1, k \in \Lambda_2, \tag{4}$$

$$x_j + y_k \leq 1 + w_{jk} \quad \forall j \in \Lambda_1, k \in \Lambda_2, \tag{5}$$

$$w_{jk} \in [0, 1] \quad \forall j \in \Lambda_1, k \in \Lambda_2. \tag{6}$$

In the second level of problem [VS] the optimal  $\bar{u} := u_2(\bar{y}) \in \mathbb{R}$  is achieved for some  $k \in \Lambda_2$ , i.e., we can write

$$\begin{aligned}
 \bar{u} &= \max_{k \in \Lambda_2} \left( \sum_{j \in \Lambda_1} U^2_{jk} \cdot x_j \right), \\
 \bar{u} &\in \mathbb{R}.
 \end{aligned} \tag{7}$$

Since only one sequence  $k \in \Lambda_2$  must be chosen, the constraint  $Fy = f$  is replaced by

$$\sum_{k \in \Lambda_2} y_k = 1. \tag{8}$$



The constraint  $Ex = e$  when written explicitly corresponds to:

$$\sum_{j \in \Lambda_1} E_{lj} \cdot x_j = 0 \quad \forall l. \tag{9}$$

We now reformulate the second level of  $[VS]$ , using a set of linear constraints. Let us denote by  $\bar{u} \in \mathbb{R}$  the maximum utility for the second player. Also, we set a large value  $M > 0$  to use the following classical big-M constraints:

$$\sum_{j \in \Lambda_1} U_{jk}^2 \cdot x_j \leq \bar{u}, \quad \forall k \in \Lambda_2, \tag{10}$$

$$\sum_{j \in \Lambda_1} U_{jk}^2 \cdot x_j \geq \bar{u} - M(1 - y_k), \quad \forall k \in \Lambda_2. \tag{11}$$

The bilevel problem  $[VS]$  is then denoted by  $[VS - L]$ , and written as follows:

$$[VS - L]: \quad u_1^{VS} = \max_{x,y,w} \sum_{j \in \Lambda_1} \sum_{k \in \Lambda_2} U_{jk}^1 \cdot w_{jk} \\ \text{s.t. (1 - 11)}$$

Note that adding a constant value to the utility function does not change the solution, therefore we can assume  $U_{jk}^2 > 1$  for all  $j \in \Lambda_1$  and  $k \in \Lambda_2$ . Under this assumption, we can add the following inequality:

$$y_k \leq \sum_{j \in \Lambda_1} U_{jk}^2 \cdot x_j \quad \forall k \in \Lambda_2, \tag{12}$$

which is valid for  $[VS]$ , since  $y_k = 0 \iff \sum_{j \in \Lambda_1} U_{jk}^2 \cdot x_j = 0$  and  $y_k = 1 \iff \exists j \in \Lambda_1$  such that  $x_j = 1$ .

Let us denote by  $[VS - L2]$  the resulting formulation:

$$[VS - L2]: \quad u_1^{VS} = \max_{x,y,w} \sum_{j \in \Lambda_1} \sum_{k \in \Lambda_2} U_{jk}^1 \cdot w_{jk} \\ \text{s.t. (1 - 12)}.$$

In what follows, we introduce a new algorithm, called  $[UBA]$  and described in Algorithm 2, that allows to compute an upper bound of the utility of the first player when a Nash equilibrium is played. We show that such upper bound is the same as the one provided by  $[VS]$ . The algorithm starts by ordering the outcomes in decreasing order of utility of the first player. Every outcome is then evaluated by solving problem  $[EC]$  with input  $\langle V_1, X_2, E \rangle$ . If the existence of a clique is proven for  $[CL]$  the algorithm stops and an upper bound is found. As remarked previously, since the procedure tests a necessary and yet not sufficient condition, the outcome found is not necessarily a realisation of a Nash equilibrium. Both  $[VS]$  and  $[UBA]$  provide the best outcome for the first player that can be a best response of a strategy of the second player, a necessary condition for the outcome to be a realisation of a Nash equilibrium. Theorem 12 thus proves that the two methods provide the same upper bound.

**Theorem 12** Consider a game  $\Gamma = \langle \mathcal{I}, \mathcal{A}, H, P, u \rangle$  and its graph form  $\langle H, E, u \rangle$ . Let  $u_1^{VS}$  and  $u_1^{UBA}$  be the optimal values obtained when  $[VS]$  and  $[UBA]$  are applied to game  $\Gamma$ , then we have  $u_1^{VS} = u_1^{UBA}$ .

**Proof** We define the set of all the outcomes that can be a best response of the second player to a strategy of the first player,  $BR_2 = \{h \in H \mid \exists s_1 \in S_1, h = \arg \max_{h' \in H(s_1)} u_2(h')\}$ . We show that  $u_1^{VS} = u_1^{UBA} = \max_{h \in BR_2} u_1(h)$ . First, we observe that  $BR_2$  corresponds to the set of feasible solutions of  $[VS]$  and thus  $u_1^{VS} = \max_{h \in BR_2} u_1(h)$ . Let us prove that  $u_1^{UBA} = \max_{h \in BR_2} u_1(h)$ . Given  $h \in H$ ,  $X = \{h' \in H \mid (h', h) \in E, u_1(h') > u_1(h)\}$  and  $V = \{h' \in H \setminus X \mid (h', h) \in E\}$ , let us consider  $H^{CL}$  the set of  $h \in H$  such that problem  $[CL]$  with input  $\langle V, H, E \rangle$  has answer true. Then we have  $u_1^{UBA} = \max_{h \in H^{CL}} u_1(h)$ . Moreover, for all  $h \in H^{CL}$  let us consider a strategy  $s_1^h \in S_1$  such that  $V \subset H(s_1^h)$  and  $H(s_1^h) \cap X = \emptyset$ . By definition of  $H^{CL}$  for each  $h \in H^{CL}$  we have  $h = \arg \max_{h' \in H(s_1^h)} u_1(h')$ , which implies  $h \in BR_2$ . Analogously, if  $h \notin H^{CL}$  there is no  $s_1 \in S_1$  such that  $h = \arg \max_{h' \in H(s_1)} u_1(h')$  and thus  $h \notin BR_2$ . Since  $H^{CL} = BR_2$ , we have  $u_1^{UBA} = \max_{h \in BR_2} u_1(h)$ .  $\square$

---

**Algorithm 2** [UBA] Upper bound algorithm

---

INPUT: Game in graph form  $\Gamma = \langle H, E, u \rangle$   
 $H^{\succeq 1} = \text{order}(H, \succeq_1)$   $\triangleright h \succeq_1 h'$  iff  $u_1(h) \geq u_1(h')$   
 $index = 0$   $\triangleright$  Start with the outcome of largest utility for player 1  
**do**  
     $h = H^{\succeq 1}(index)$   $\triangleright$  Select the outcome corresponding to  $index$   
     $u_1^{UBA} = u_1(h)$   
     $V_1 = \{h' \in H \setminus X_2 \mid (h', h) \in E\}$   $\triangleright$  vertex set for player 1  
     $X_2 = \{h'' \in H \mid (h'', h) \in E, u_2(h'') > u_2(h)\}$   $\triangleright$  excluding set for player 1  
    **solve**  $[CL]$  giving  $\langle V_1, X_2, E \rangle$  as input  
     $index = index + 1$   
**while**  $[CL]$  has no solution  
OUTPUT:  $u_1^{UBA}$

---

### 4.3 Best and Worst Nash Equilibrium

Note that the algorithms  $[VS]$  and  $[UBA]$  introduced in Sect. 2.2 do not provide the tightest upper bound to the utility of the Nash equilibria (cf. Appendix A.6). In this section, we introduce an algorithm that provides a Nash equilibrium outcome whose utility is the highest for the first player. The algorithm  $[BNE]$  consists in ordering the outcomes from best to worst with respect to the utility function of the first player and then picking the first of them that is a realisation of a Nash equilibrium, i.e., for which  $[NE]$  admits a feasible solution. Analogously, it is possible to identify the Nash equilibrium with the lowest utility for the first player, by ordering in reverse order the outcomes. The algorithm  $[WNE]$  is presented together with  $[BNE]$  in Appendix B.3.

## 5 Numerical Results

To our knowledge, the only method for enumerating the pure Nash equilibria of extensive-form games is a brute-force transformation of the game in normal form. This approach requires an enumeration of the exponential number of strategy profiles of the game, which cannot be tested numerically, unless we limited the size of games to few (less than 20) outcomes.

For this reason, in this section, we assess the performance of the methods introduced in Sect. 4 through several series of experiments. To our knowledge, there is no standard library of extensive-form games in the literature. For this reason, we developed a new library presented in Sect. 5.1. The methods that provide bounds to the utility of Nash equilibria and those who enumerate them are analysed separately, respectively in Sect. 5.2 and in Sect. 5.3. The experimental study was conducted on a Intel Xeon CPU 2.20 GHz with 13 GB RAM. The algorithms were implemented in Python 3.8. The solver used to solve all Mixed-Integer Linear Programming problems is GLPK (cf. [30]).

## 5.1 Library of Extensive-Form Games

Extensive-form games with perfect recall and perfect information have never been categorised. We thus introduce a new classification of games based on three key features: the *structure* of the game-tree, the *size* of the game and the *utility* function. This allows to challenge our algorithms on a wide range of game instances and analyse their efficiency. The proposed classification will be used to create a new library of extensive-form games. More precisely, the structure of a game captures the properties of the shape of the game-tree. The size of the game (i.e., the number of outcomes) allows us to better assess the scalability of methods. Finally, once structure and size are fixed, the only parameter that varies in a game is the utility function, which we provide different families of functions for.

Each instance of the dataset is referred to with a specific name encoding the three key features of the game. Such encoding is shown in Table 1. The games' structure is encoded as follows:  $Rn$  indicates that the number of actions at every child of a node  $h' \in H$  is chosen uniformly at random  $\mathbb{U}(\{0, \dots, n \cdot |\mathcal{A}(h')\})$ , given the constraint that they have on average  $n$  actions;  $Cn$  indicates that every node has the same number of actions, with  $n$  actions per node; and finally  $Un$  indicates that, at every node, all actions but one lead to an outcome, with  $n$  actions per node. The players' utility is encoded as follows:  $R$  if the utility of an outcome is chosen uniformly at random  $\mathbb{U}([0, 1])$  in the interval  $[0, 1]$ ;  $D$  if the utility is chosen randomly from a discrete set  $\mathbb{U}(\{1, \dots, 10\})$ , namely a natural integer between 1 and 10;  $Z$  if the game is zero-sum, i.e., at every node  $h \in H$  the winner is chosen at random  $i \in \{1, 2\}$  and gets  $u_i(h) = 1$ , while the loser gets 0;  $A$  if the game is zero-sum and the winner has a utility chosen randomly from a discrete set  $\mathbb{U}(\{1, \dots, 10\})$ ;  $F$  if the utility is chosen randomly from  $\mathbb{U}([0, 1])$  at every outcome and it is the same for the two players;  $E$  if every utility of every outcome has constant value. The latest value of a label is the size of the game.

**Example** An instance of game labeled C4E100 has 100 outcomes, every node has 4 actions ( $C$ ) and utility is a constant function ( $E$ ).

The library is publicly accessible (cf. [31]) and composed of three distinct datasets. Dataset 1 contains 21 extensive-form games which vary in their structure, within the range  $\{R, C, U\}$ , and in their size  $\{100, 216, 324, 400, 512, 625, 729\}$ , but not in their utility, which is always Random  $R$ . Dataset 1 has games of smaller size, i.e., small enough to manage methods already known in the literature, that provide bounds to the utility of Nash equilibria; Dataset 1 is used in Sect. 5.2. Dataset 2 has 72 extensive-form games which vary in their structure  $\{R, C, U\}$ , size  $\{256, 729, 1296, 2401\}$  and utility  $\{R, D, Z, A, F, E\}$ . Dataset 3 has 75 extensive-form games of size 729 which vary in structures  $\{R, C, U\}$  and utility  $\{R, D, Z, A, F\}$ . Dataset 2 and 3 have games of larger size, that are used to assess the method to enumerate the Nash equilibria outcomes. They are used in Sect. 5.3. Dataset 2 includes

**Table 1** Label encoding of the games

Label	Structure	Definition
R	Random	Every node has a random number of actions
C	Complete	Every node has the same number of actions
U	Totally unbalanced	At every node all children but one are outcomes
Label	Utility	Definition
R	Random	$u_i(h) \sim \mathbb{U}(0, 1), \forall i \in \{1, 2\}, \forall h \in H$
D	Discrete	$u_i(h) \sim \mathbb{U}(\{1, 2, 3, \dots, 10\}), \forall i \in \{1, 2\}, \forall h \in H$
Z	Zero-sum	$u_i(h) = 1, u_j(h) = 0, i \sim \mathbb{U}(\{1, 2\}), i \neq j, \forall h \in H$
A	Asymmetric	$u_i(h) \sim \mathbb{U}(\{1, 2, 3, \dots, 10\}), u_j(h) = 0, i \sim \mathbb{U}(\{1, 2\}), i \neq j, \forall h \in H$
F	Indifferent	$u_1(h) = u_2(h) \sim \mathbb{U}(0, 1), \forall h \in H$
E	Equal	$u_i(h) = 1, \forall i \in \{1, 2\}, \forall h \in H$

Upper table: coding of the structure of the tree. Lower table: coding of the utility function

games with different characteristics; it enables us to assess the method on a large variety of games. Dataset 3 includes 75 games with size 729, gathered in 15 groups, each one with 5 games having the same encoding. The 15 groups are built alternating 3 different structures  $\{R3 * 729, C3 * 729, U5 * 729\}$  and 5 different utilities  $' *' \in \{R, D, Z, A, F\}$ . The utility  $E$  has been discarded, as we show next that it does not need further analysis. The analysis performed on Dataset 3 allows to understand the variability of the results obtained for Dataset 2.

## 5.2 Bounds to the Utility of Nash Equilibria

We first test the methods introduced in Sect. 4.2 and in Sect. 4.3 on Dataset 1. The reference methods known in literature, i.e., the branch and bound algorithms used to solve formulations  $[VS - L]$  and  $[VS - L2]$ , are compared to algorithm  $[UBA]$ : they all provide the same upper bound to the utility of the first player in a Nash equilibrium. We recall that the proposed algorithms  $[BNE]$  and  $[WNE]$  provide the tightest (respectively, upper and lower) bounds to the utility of every Nash equilibrium. Table 2 reports the computation time for every instance of each algorithm as well as the number of iterations for the three iterative algorithms  $[UBA]$ ,  $[BNE]$  and  $[WNE]$ .

*Comparison of  $[UBA]$  vs  $[VS - L]$  vs  $[VS - L2]$ .* First and main result is that the algorithm  $[UBA]$  overcomes the methods relying on  $[VS - L]$  and  $[VS - L2]$  by some orders of magnitude. The method  $[UBA]$  computes the upper bound to the utility of Nash equilibria at least  $> 40$  times faster than  $[VS - L]$  and  $[VS - L2]$ . Second, we highlight the quality of the valid inequality added in  $[VS - L2]$ . Indeed, we observe that it improves the computation time with respect to  $[VS - L]$  in 10 out of 21 instances and reduces of around 30% the computation time in instances like  $R6R512$  and  $C2R512$ , which require more than 25 minutes for  $[VS - L]$ . Both  $[VS - L]$  and  $[VS - L2]$  perform 17% to more than 99% faster on structures  $C$  and  $U$  than those of type  $R$ ; we ascribe this to the fact that these games present fewer nodes and thus require fewer constraints in the linear formulation.

*Comparison of  $[UBA]$  vs  $[BNE]$  vs  $[WNE]$ .* Algorithm  $[BNE]$  allows us to tighten the upper bound in 4 instances out of 21, thus showing that  $[UBA]$ ,  $[VS - L]$  and  $[VS - L2]$  do not always achieve the tightest bound. In Table 2, the few instances in which  $[BNE]$  tightens the bound are underlined, whereas in the other 17 out of 21 instances the number of iterations does not change. As predicted,  $[BNE]$  is slower than  $[UBA]$ , but still performs more than 6 times faster than both  $[VS - L]$  and  $[VS - L2]$  in average on this dataset. Algorithm  $[WNE]$  requires a larger computation time than  $[BNE]$  in 19 out of 21 instances. In fact, it always has a larger number of iterations than  $[BNE]$ . Indeed,  $[WNE]$  first checks the outcomes with lowest utility for the first player, those are unlikely to correspond to the best response of the first player. It takes thus far more time to identify an outcome which is a best response for both players.

## 5.3 Enumeration of Realisations of Nash Equilibria

In the next experiment, we tested the methods introduced in Sect. 4.1 on Dataset 2 and on Dataset 3. We have measured the performance of Algorithm  $[EA]$  while enumerating the Nash equilibria outcomes of every instance.

*Analysis on Dataset 2.* For the space's sake, in Table 3 we display only the results for all the games' instances with size 729. More precisely, for each game, we show how many outcomes are the realisations of Nash equilibria  $NE$ , the average size  $X_{avg}$  of all sets  $X_1$  and

**Table 2** Comparison of the CPU time for computation of algorithms providing bounds to the utility of NE

Game label	Size	Structure	Computation time (s)			Iterations				
			[VS-L]	[VS-L2]	[UBA]	[BNE]	[WNE]	[UBA]	[BNE]	[WNE]
R4R100	100	R	4.85	3.69	0.09	0.14	0.98	12	12	69
C10R100	100	C	0.32	0.46	< 0.01	< 0.01	0.10	1	1	30
U4R100	100	U	2.22	2.75	0.02	0.18	0.57	29	29	35
R4R216	216	R	51.34	44.23	< 0.01	< 0.01	1.53	2	2	124
C6R216	216	C	10.45	22.87	0.04	0.38	9.97	6	7	164
U6R216	216	U	19.43	12.03	0.06	0.61	2.24	39	39	178
R5R324	324	R	324.76	282.29	0.04	2.93	9.35	5	5	287
C18R324	324	C	2.11	2.97	0.01	0.28	0.22	5	5	57
U18R324	324	U	26.42	25.37	0.03	1.48	5.22	13	13	312
R5R400	400	R	512.75	461.24	2.20	2.71	9.49	32	32	334
C20R400	400	C	2.83	2.83	0.03	0.46	0.27	12	12	69
U4R400	400	U	234.33	248.02	0.27	2.21	5.68	60	60	341
R6R512	512	R	1918.39	1469.56	0.53	4.12	26.97	47	78	428
C2R512	512	C	1594.47	1114.76	0.25	0.38	161.49	1	1	317
U8R512	512	U	192.89	176.96	0.27	3.50	16.66	63	89	424
R6R625	625	R	> 1 h	> 1 h	0.11	7.18	24.97	15	15	558
C5R625	625	C	95.0	95.92	0.30	2.08	7.77	22	22	182
U4R625	625	U	818.2	854.45	0.04	7.94	29.96	4	4	622
R7R729	729	R	> 1 h	> 1 h	11.13	9.31	24.71	56	56	419
C3R729	729	C	104.33	126.89	0.02	3.02	32.75	3	3	244
U14R729	729	U	255.96	247.0	0.17	15.19	29.92	30	45	685

Every line is an instance. Running time is measured in seconds. Time limit is set to 1 h (> 1 h indicates that such limit is reached). The instances in which [BNE] tightens the bound of [UBA] are underlined

**Table 3** Application of algorithm [EA] on games of dataset 2 with size 729

Label	Structure	Utility	NE	$X_{avg}$	$t_{tot}$	$t_{avg}$	$t_{max}$
R3R729	R	R	4	182.0	115.99	0.16	34.46
R3D729		D	18	163.8	31.92	0.04	7.91
R3Z729		Z	4	91.0	10.06	0.01	2.24
R3A729		A	8	133.4	100.32	0.14	6.63
R3F729		F	12	182.0	83.15	0.11	8.23
R3E729		E	729	0.0	0.48	< 0.01	< 0.01
C3R729	C	R	76	182.0	881.46	1.21	4.76
C3D729		D	98	163.7	999.99	1.37	18.79
C3Z729		Z	271	91.1	466.35	0.64	3.64
C3A729		A	90	139.1	1297.1	1.78	9.10
C3F729		F	133	182.0	1039.19	1.43	6.75
C3E729		E	729	0.0	0.46	< 0.01	< 0.01
U5R729	U	R	3	182.0	43.63	0.06	8.55
U5D729		D	7	163.9	20.99	0.03	7.24
U5Z729		Z	1	91.0	3.78	0.01	0.02
U5A729		A	1	137.1	44.55	0.06	8.75
U5F729		F	8	182.0	39.94	0.05	7.67
U5E729		E	729	0.0	0.48	< 0.01	< 0.01

Every line is an instance. Computation time is in seconds

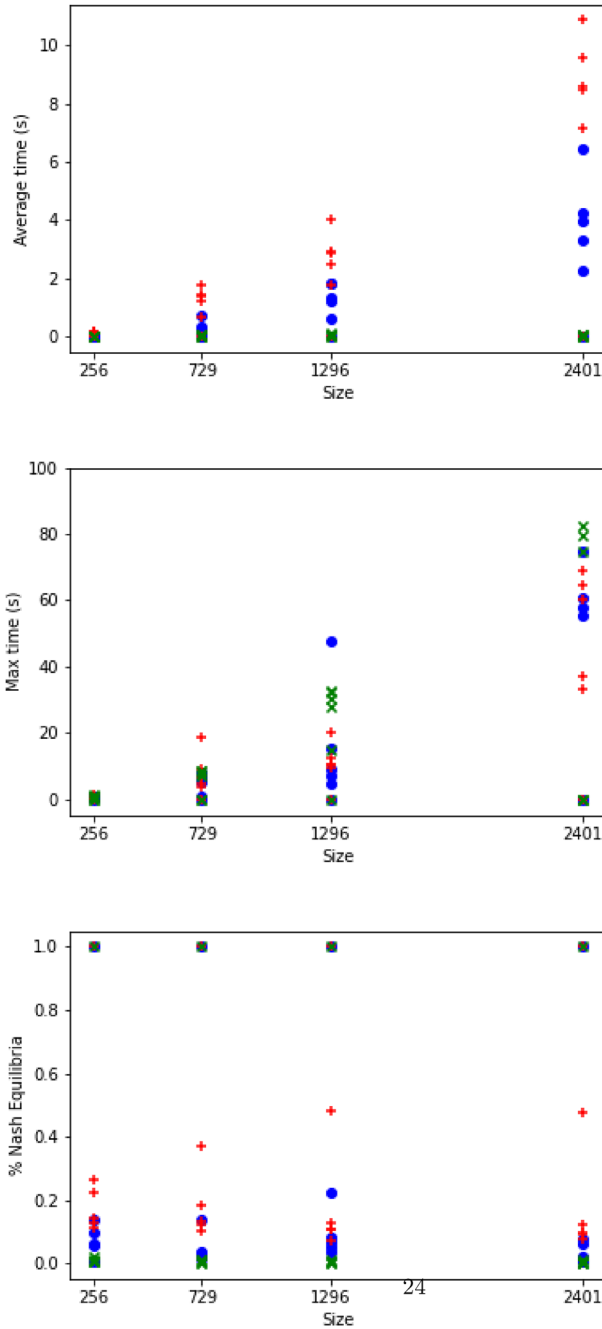
$X_2$ , the total time required to run the algorithm  $t_{tot}$ , the average time  $t_{avg}$  and the maximal time  $t_{max}$  for an outcome of the game. In addition, Fig. 4 displays the average time and the maximal time required to execute the algorithm on a game outcome, and the percentage of Nash equilibria identified among all the game's outcomes.

*Impact of the Size on Performance.* As expected, the computation time increases with the size of the game (cf. Fig. 4). Algorithm [EA] iterates over the outcomes and requires to solve system [NE] for each outcome. When the size of the game increases, on average the size of system [NE] increases accordingly. However, if the size is fixed, the structure and the utility function have a fundamental impact on the computation time.

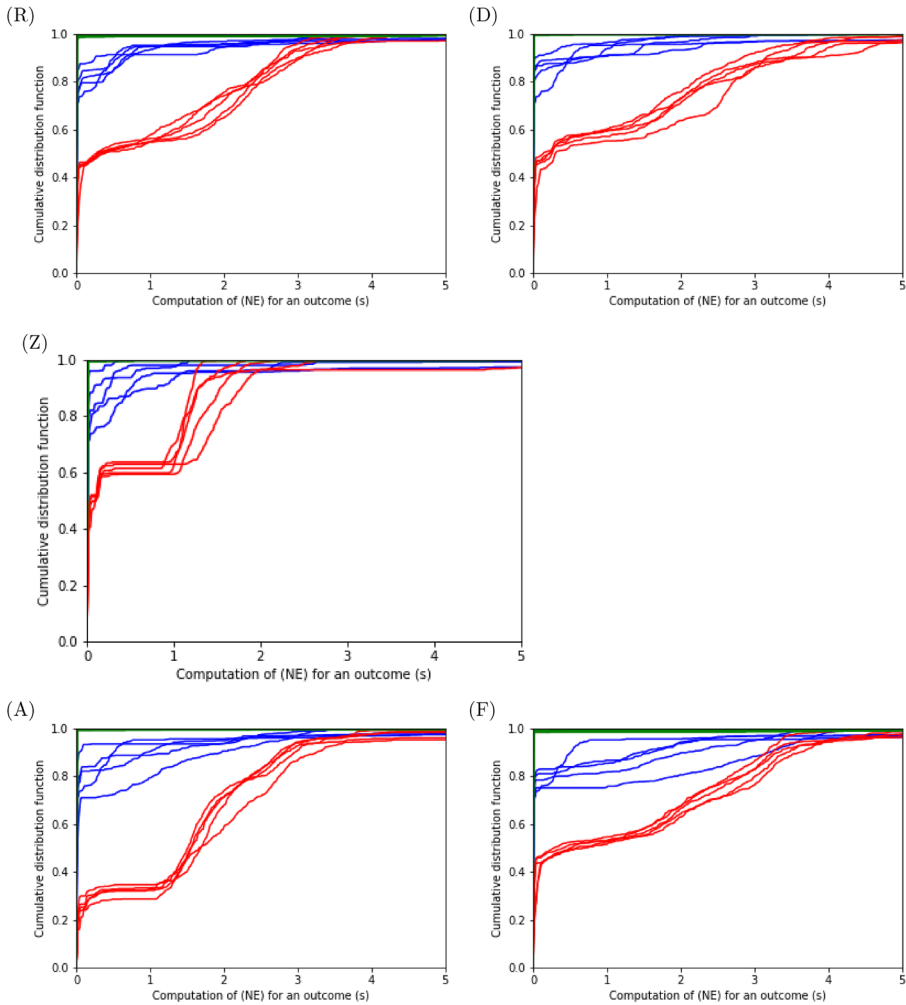
*Impact of the Utility Function on Performance.* In the degenerate case  $E$ , when outcomes all have same utility value, solving system [NE] gets trivial for each outcome and thus the computation time of each iteration is negligible (cf. Table 3). Indeed, the sets of outcomes to be excluded  $X_1$  and  $X_2$  are empty at every iteration. For such case building the graph is not even necessary. We observe that the computation time is lower for the case  $Z$ , as the sizes of  $X_1$  and  $X_2$  are smaller. We cannot infer significant correlations on the other cases ( $R$ ,  $D$ ,  $A$ ,  $F$ ) and we thus defer such analysis to Dataset 3.

*Impact of the Structure on Performance.* The structure of the game influences the average time necessary to verify if an outcome is the realisation of a Nash equilibrium. Indeed, we observe in Fig. 4 that in games with the same size those whose nodes have the same number of children (structure coding  $C$ ) require more time on average to compute an equilibrium. This is due to the fact that the neighborhoods of an outcome  $V_1 \cup X_2$  and  $V_2 \cup X_1$  in the graph and its complementary have always the same size in both parts of system [NE]. On the other hand, in games with more asymmetrical structure (structure coding  $R$  and  $U$ ) one of





**Fig. 4** Application of algorithm [EA] to dataset 2. Times are in seconds. Structure of the game is identified by colour and shape: blue circle (Random), red plus (Complete), green cross (totally Unbalanced) (Color figure online)



**Fig. 5** Computation time distribution for  $[NE]$  on the outcomes of games belonging to Dataset 3. Every curve corresponds to the distribution for the outcomes of a single game. Each figure reports on the performance on the 15 games having the same encoding for the utility function, being respectively Random (R), Discrete (D), Zero-sum (Z), Asymmetric (A) and Indifferent (F). The distribution curves represent each a structure: Random (R) in blue, Complete (C) in red and Totally Unbalanced (U) in green (Color figure online)

the two graphs is often smaller and thus the corresponding  $[NE]$  is much faster to be solved in practice.

*Analysis of Dataset 3.* Table 4 shows the average of performances on 5 sample games for each one of the 15 patterns. The outcomes are chosen with uniform random probability. Every curve of Fig. 5 shows the computation time distribution of  $[NE]$  for an outcome of a game. The 15 curves provide such result for the 15 games having the same utility coding. Namely, every line appearing in the plots shows for every game type  $\Gamma = \langle \mathcal{I}, H, u \rangle$  the function  $f(t) = \mathbb{P}(t(h) \leq t | h \in H)$ , i.e., the probability that the algorithm solving system  $[NE]$  takes less than  $t$  seconds to determine if  $h$  is the realisation of a Nash Equilibrium.

**Table 4** Application of algorithms of [EA] to dataset 3

Label	Structure	Utility	NE	$t_{tot}$	$t_{avg}$	$t_{max}$
R3R729	R	R	17.2	206.61	0.28	7.11
C3R729	C		75.2	883.40	1.21	28.89
U5R729	U		2.0	47.42	0.06	9.72
R3D729	R	D	38.6	149.10	0.20	5.39
C3D729	C		101.8	1240.58	1.70	69.77
U5D729	U		8.0	12.96	0.02	4.37
R3Z729	R	Z	31.4	70.24	0.10	4.01
C3Z729	C		276.6	654.16	0.90	26.68
U5Z729	U		1.8	4.34	0.01	0.02
R3A729	R	A	10.0	250.22	0.34	4.96
C3A729	C		95.4	1523.49	2.10	167.65
U5A729	U		1.8	40.07	0.06	7.74
R3F729	R	F	91.4	364.85	0.50	6.86
C3F729	C		132.8	1354.45	1.86	121.96
U5F729	U		8.8	58.53	0.08	7.72

Every line shows the average value of 5 instances having the same label. All games have size 729. Computation time is in seconds

*Impact of the Structure on Performance.* These numerical results show empirically that the structure is the main factor influencing the performance of the algorithm solving system [NE]. Indeed, on a game with totally unbalanced (U) structure it takes a negligible time for the vast majority (around 99%) of the outcomes of a game instance (cf. Fig. 5). On the other hand, on a game with complete (C) structure the algorithm such percentage decreases to 20 – 50%. The algorithm performs in between the two extremes for a game with random (R) structure.

*Impact of the Utility on Performance.* We observe on Table 4 that the best performances are obtained for zero-sum games (coding utility Z) for the same structure. This is due to the fact that when the utility is equal to 1 the set of outcomes to be excluded X is empty (cf. Table 3). One of the two sides of the problem is always trivial. If the game switches from utility Z to A, i.e., if the player that wins gains a value between 1 and 10 instead of just 1, the property that makes one of the problems trivial is lost. Indeed, games with coding A have performances comparable to those with other utility codings (R, D, F). One might thus assume that increasing the granularity of the utility might make the algorithm less efficient. However, we do not observe any significant difference in the computation time while comparing (D) ( $u_i \sim \mathbb{U}(\{1, \dots, 10\})$ ) to (R) ( $u_i \sim \mathbb{U}(0, 1)$ ).

### 5.4 Obtaining Insights on Nash Equilibria of Extensive-Form Games

To the best of our knowledge, Algorithm [EA] is the first one proposed to enumerate the Nash equilibria outcomes of an extensive-form game that does not resort to brute force. Besides the analysis of its complexity, we can also provide further insights of the numerosity of Nash equilibria in a game.

The number of possible realisations of a Nash equilibrium can vary from 1 to the number of outcomes of a game. The latter case implies that the utility function is constant (case E).

However, in the generic case the number of Nash equilibria highly depends on the structure of the game-tree. We observe from Sect. 5.3 that games with a totally unbalanced structure ( $U$ ) tend to host fewer Nash equilibria. This is due to the fact that if there is an outcome with great value to a player, she might choose to stick to it, and the opponent has few options to build a strategy that redirects her to an outcome that lies deeper in the tree. If an outcome with great value happens to lie at a very high level of the tree, the lower levels hardly host other realisations of Nash equilibria. The converse tends to occur as well, i.e., in a generic game with a complete structure ( $C$ ) we typically observe many more Nash equilibria than in the unbalanced case. This follows the intuition that in a complete game structure a player can find more combination of strategies to convince the opponent to switch her best response to a different outcome.

## 6 Conclusions

In this paper, we introduce a new representation for a two-player extensive-form game with perfect information as a graph of its outcomes. We prove that identifying a Nash equilibrium outcome of an extensive-form game corresponds to identifying two cliques on such graph and its complementary. Thanks to this result, we introduce the first method of the literature to determine if an outcome of an extensive-form game is a realisation of a Nash equilibrium. Such method allows to define the first algorithm to enumerate the realisations of all Nash equilibria of an extensive-form game. The algorithm performs very well on a sample dataset of games of different structures and sizes. Moreover, it is possible to reframe the algorithm so that it provides any given bound to the utility of all Nash equilibria. Such algorithm performs significantly better than the most known method in literature, which provides a (not always the tightest) upper bound to the utility of Nash equilibria.

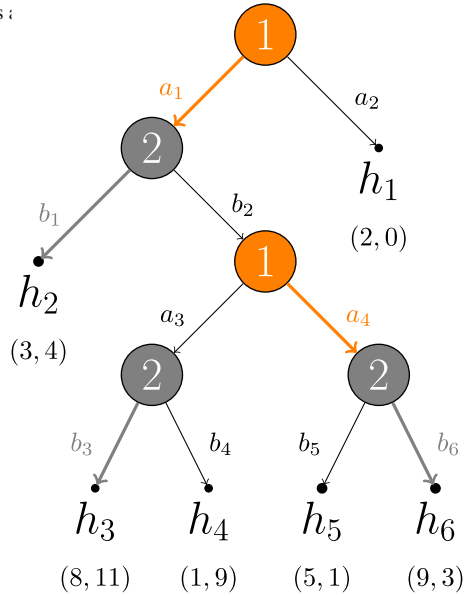
We foresee several extensions for this work. First, we do not fully exploit the properties of the graph form to improve the efficiency of the proposed algorithms. Second, it would be interesting to devise methods to parallelize the computation of the Nash equilibria outcomes in large instances. Finally, the categorisation of extensive-form games suggests that more efficient methods can be designed for some specific classes of games. For instance, customized algorithms can be written to compute the compatibility of two outcomes. A possible extension of this results would be eventually to extend the numerical results to larger games with mixed strategies, with more than two players and with imperfect information.

## Appendix A: Examples

### A.1 Extensive Form

We consider an example of extensive-form game, whose representation is the tree of Fig. 6. The game  $\Gamma = \langle \mathcal{I}, H, u \rangle$  with two players  $\mathcal{I} = \{1, 2\}$  is represented by a tree, which allows to show all the possible actions. The first player observes no actions  $\emptyset$  and she picks an action in the set  $\mathcal{A}(\emptyset) = \{a_1, a_2\}$ . If she picks action  $a_2$ , the game arrives to the outcome  $h_1 = (a_2) \in H$ . On the other hand, if she picks action  $a_1$ , the second player observes it and picks an action from the set  $\mathcal{A}(a_1) = \{b_1, b_2\}$ . If the second player picks action  $b_1$ , she gets to the outcome  $h_2 = (a_1, b_1) \in H$ . Analogously it is possible to get all the other outcomes of the game:  $h_3 = (a_1, b_2, a_3, b_3)$ ,  $h_4 = (a_1, b_2, a_3, b_4)$ ,  $h_5 = (a_1, b_2, a_4, b_5)$ ,  $h_6 =$

**Fig. 6** An extensive-form game in its representation as : Preferences. The order of preference for the first player is:  $h_6 \succ_1 h_3 \succ_1 h_5 \succ_1 h_2 \succ_1 h_1 \succ_1 h_4$ . The order of preference for the second player is:  $h_3 \succ_2 h_4 \succ_2 h_2 \succ_2 h_6 \succ_2 h_5 \succ_2 h_1$ . *Strategies.* The two players choose respectively  $s_1$  and  $s_2$  as strategies. At nodes  $\emptyset$  and  $(a_1, b_2)$  the first player chooses respectively  $s_1(\emptyset) = a_1$  and  $s_1(a_1, b_2) = a_4$ . At nodes  $(a_1)$ ,  $(a_1, b_2, a_3)$  and  $(a_1, b_2, a_4)$  the second player chooses respectively  $s_2(a_1) = b_1$ ,  $s_2(a_1, b_2, a_3) = b_3$  and  $s_2(a_1, b_2, a_4) = b_6$



$(a_1, b_2, a_4, b_6)$ . The set of outcomes is therefore  $H = \{h_1, h_2, h_3, h_4, h_5, h_6\}$ . The set of histories is instead  $H' = \{\emptyset, h_1, (a_1), h_2, (a_1, b_2), (a_1, b_2, a_3), (a_1, b_2, a_4), h_3, h_4, h_5, h_6\}$ . The function  $P$  maps respectively  $H' \setminus H = \{\emptyset, (a_1), (a_1, b_2), (a_1, b_2, a_3), (a_1, b_2, a_4)\}$  to the players acting at such nodes  $\{1, 2, 1, 2, 2\}$ . The utility function  $u : H \rightarrow \mathbb{R}^2$  evaluates the outcomes. For instance, we can have  $u(h_1) = (2, 0)$ , where  $u_1(h_1) = 2$  is the evaluation given to  $h_1$  by the first player and  $u_2(h_1) = 0$  is the one given by the second player. For the game in Fig. 6 we assume that  $u(h_2) = (3, 4)$ ,  $u(h_3) = (8, 11)$ ,  $u(h_4) = (1, 9)$ ,  $u(h_5) = (5, 1)$ ,  $u(h_6) = (9, 3)$ . The value of the utility allows to understand which outcomes are preferred by the players. For instance, the second player prefers  $h_3$  to  $h_6$  because  $u_2(h_3) = 11 > 3 = u_2(h_6)$ . We write alternatively  $h \succ_i h'$  to show that  $u_i(h) > u_i(h')$ . In the caption of Fig. 6 all the preferences among the outcomes are given, based on the values assigned by the utility function.

*Strategies.* In the game of Fig. 6 the first player acts at two different moments in which she observes respectively  $\emptyset$  and  $(a_1, b_2)$ : we have thus  $H_1 = \{\emptyset, (a_1, b_2)\}$ . According to Definition 2, a strategy for player 1 is a function that maps every history observed in  $H_1$  to an action that can be chosen by her. In Fig. 6 the choices of the players are marked by thicker arrows. In the example, the first player picks strategy  $s_1$ , where  $s_1(\emptyset) = a_1$  and  $s_1(a_1, b_2) = a_4$ . With some abuse of notation, we write  $s_1(a_1, b_2)$  instead of  $s_1((a_1, b_2))$  to simplify the notation. The histories observed by the second player are  $H_2 = \{(a_1), (a_1, b_2, a_3), (a_1, b_2, a_4)\}$ . In Fig. 6 the second player picks strategy  $s_2$ , where:  $s_2(a_1) = b_1$ ;  $s_2(a_1, b_2, a_3) = b_3$ ;  $s_2(a_1, b_2, a_4) = b_6$ .

*Application of Theorem 1.* We apply Theorem 1 to the game of Fig. 6. The set of the outcomes of Nash equilibria is  $\{h_2, h_3\}$ . If we compute the solution of [ST] for such game, we get  $(\bar{s}_1, \bar{s}_2) \in S_1 \times S_2$ , where  $\bar{s}_1(\emptyset) = a_1$ ,  $\bar{s}_1(a_1, b_2) = a_3$ ,  $\bar{s}_2(a_1) = b_2$  and  $\bar{s}_2(a_1, b_2, a_3) = b_3$ . The realisation of the strategy profile  $(\bar{s}_1, \bar{s}_2)$  is  $h_3$ . Coherently with Theorem 1, the solution  $u_1(h_3)$  is, indeed, an upper bound for the utility of every outcome in the set of Nash equilibria  $\{h_2, h_3\}$ .

*Sequences.* We consider the framework introduced in Sect. 4.2. The sequences of the first player are  $\Lambda_1 = \{\emptyset, (a_1), (a_2), (a_1, a_3), (a_1, a_4)\}$ , while the sequences of the second player are  $\Lambda_2 = \{\emptyset, (b_1), (b_2), (b_2, b_3), (b_2, b_4), (b_2, b_5), (b_2, b_6)\}$ . If the first player chooses action  $a_1$ , she must choose either action  $a_3$  or  $a_4$ . Therefore the constraint  $x_{(a_1)} = x_{(a_1, a_3)} + x_{(a_1, a_4)}$  is added to the formulation. Similarly, if the second player chooses action  $b_2$ , she must choose at least one action among  $\{b_3, b_4, b_5, b_6\}$ . Therefore we add the constraint  $y_{(b_2)} = y_{(b_2, b_3)} + y_{(b_2, b_4)} + y_{(b_2, b_5)} + y_{(b_2, b_6)}$ .

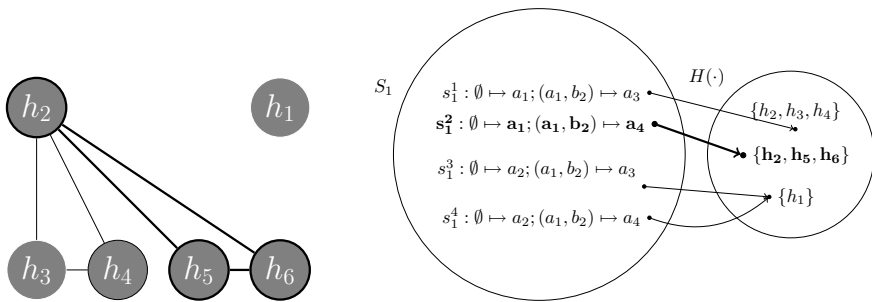
### A.2 Graph Form

We take into account the game of Fig. 6 with its corresponding graph form in Fig. 7a. Let us show, for instance, that outcomes  $h_2$  and  $h_4$  are compatible. We define:  $s_1 \in S_1$  such that  $s_1(\emptyset) = a_1$  and  $s_1(a_1, b_2) = a_3$ ;  $s_2 \in S_2$  such that  $s_2(a_1) = b_1$ ;  $s'_2 \in S_2$  such that  $s'_2(a_1) = b_2$  and  $s'_2(a_1, b_2, a_3) = b_4$ . We have that  $(s_1, s_2) \mapsto h_2$  and  $(s_1, s'_2) \mapsto h_4$ : they are therefore compatible. On the other hand there is no strategy of the first player that can lead to both  $h_2$  and  $h_1$ , because different actions would occur at the beginning of the game, i.e., at node  $\emptyset$ . With similar arguments, it is possible to build all the graph form of the game.

*Strategies.* We consider the game of Fig. 6 and a strategy  $s_1 \in S_1$  such that  $s_1(\emptyset) = a_1$  and  $s_1(a_1, b_2) = a_4$ . By definition we have that  $H(s_1) = \{h_2, h_5, h_6\}$ , which induces indeed a maximal clique in the graph of Fig. 7a. Four different strategies are available in  $S_1$  to the first player, since we have two different actions that can be played for every node  $\emptyset$  and  $(a_1, b_2)$ . There are 3 maximal cliques in graph  $\langle H, E \rangle$ , i.e., those induced by the sets of outcomes  $\{h_1\}$ ,  $\{h_2, h_3, h_4\}$  and  $\{h_2, h_5, h_6\}$ , respectively. This small example shows that the graph form is a less redundant representation of the strategies of the players than the normal form. In fact, two different strategies induce the set of outcomes  $\{h_1\}$ .

### A.3 Application of Theorem 7

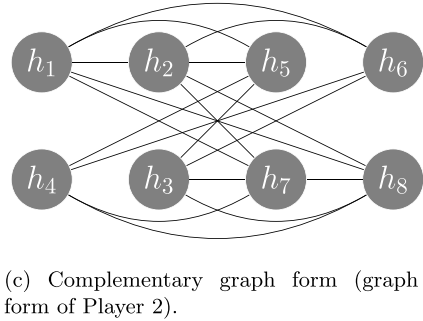
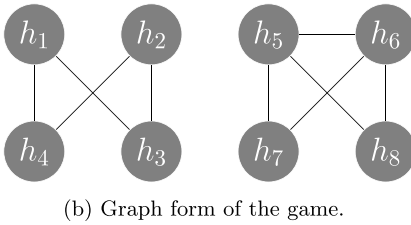
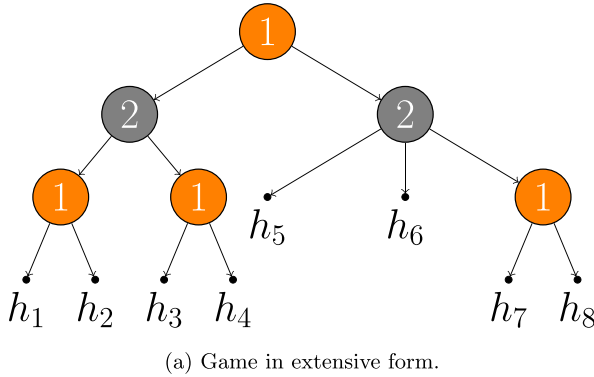
We consider the game in extensive form of Fig. 8a game which can be solved by direct inspection of the graph. Indeed, let us enumerate the realisations of Nash equilibria of the game by applying Theorem 7 to every outcome. In this example, the preferences of the players over the outcomes are respectively  $u_1 : h_3 >_1 h_4 >_1 h_5 >_1 h_6 >_1 h_8 >_1 h_7 >_1 h_2 >_1 h_1$  and  $u_2 : h_1 >_2 h_2 >_2 h_4 >_2 h_8 >_2 h_5 >_2 h_6 >_2 h_7 >_2 h_3$ . In order to apply Theorem 7 we



(a) Graph form representation of the game of Figure A1.

(b) Every strategy for the first player induces a maximal clique of the graph.

Fig. 7 Strategies as maximal cliques



**Fig. 8** Example. Preferences of the players over the outcomes are respectively:  $u_1 : h_3 \succ_1 h_4 \succ_1 h_5 \succ_1 h_6 \succ_1 h_8 \succ_1 h_7 \succ_1 h_2 \succ_1 h_1$  and  $u_2 : h_1 \succ_2 h_2 \succ_2 h_4 \succ_2 h_8 \succ_2 h_5 \succ_2 h_6 \succ_2 h_7 \succ_2 h_3$

consider the graph  $\langle H, E \rangle$  of the game depicted in Fig. 8b. By directly inspecting this graph and its complementary  $\langle H, E^C \rangle$  (cf. Fig. 8c), we achieve the following results.

- $h_1$  is not an outcome of a Nash equilibrium:  $h_1$  is the outcome least preferred by the first player. There is no strategy of the second player, i.e., there is no maximal clique on the complementary graph  $\langle H, E^C \rangle$  which includes  $h_1$  alone;
- $h_2$  is not an outcome of a Nash equilibrium: the only maximal cliques on the complementary graph including  $h_2$  are  $\{h_1, h_2, h_5\}$ ,  $\{h_1, h_2, h_6\}$  and  $\{h_1, h_2, h_7, h_8\}$ , but in none of these cases  $h_2$  is the favourite outcome by the first player. This means that there is no strategy of the second player such that  $h_2$  corresponds to the outcome of the best response of the first player;
- $h_3$  is not an outcome of a Nash equilibrium: it is the least preferred outcome by the second player. Any maximal clique on the complementary graph which includes  $h_3$  also includes another outcome which is preferred by the second player to  $h_3$ ;
- $h_4$  is not an outcome of a Nash equilibrium: every maximal clique on the complementary graph which includes  $h_4$  also includes  $h_3$ , which is preferred by the first player to  $h_4$ ;
- $h_5$  is a possible outcome of a Nash equilibrium: we identify the maximal cliques  $\{h_5, h_6, h_7\}$  and  $\{h_1, h_2, h_5\}$  respectively on the graph form and its complementary. In both cases  $h_5$  is the preferred outcome, respectively by the second and the first player;
- $h_6$  is not an outcome of a Nash equilibrium: every maximal clique in  $\langle H, E \rangle$  which includes  $h_6$  also includes  $h_5$ , which is preferred by the second player to  $h_6$ ;



- $h_7$  is not an outcome of a Nash equilibrium: every maximal clique on  $\langle H, E^C \rangle$  which includes  $h_7$  also includes  $h_8$ , which is preferred by the first player to  $h_7$ ;
- $h_8$  is a possible outcome of a Nash equilibrium: we identify the maximal cliques  $\{h_5, h_6, h_8\}$  and  $\{h_1, h_2, h_7, h_8\}$  respectively on the graph  $\langle H, E \rangle$  and its complementary  $\langle H, E^C \rangle$ . In both cases  $h_8$  is the preferred outcome, respectively by the second and the first player.

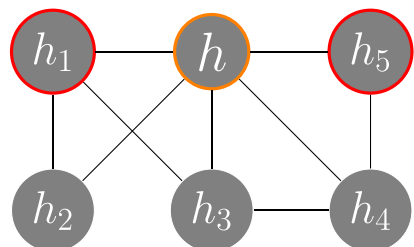
The only two outcomes which are possible realisations of Nash equilibria are thus  $h_5$  and  $h_8$ . It is interesting to notice that they are not Pareto-optimal, since  $h_4$  is preferred by both players. The Subgame Perfect Equilibrium of a game is also a Nash equilibrium, therefore it must be either of the two outcomes. Indeed, by backward induction it is proven to be  $h_8$ .

### A.4 Formulation of [CL]

Let us consider the graph  $\langle H, E \rangle$  of Fig. 9 and evaluate  $h$  as a possible realisation of a Nash equilibrium. We would like to answer if  $h$  is the best response for a player  $i$  (it is not necessary to specify which player in this mock example), i.e.  $h$  to be the best outcome of a strategy. We assume given that  $h_1 \succ_i h$  and  $h_5 \succ_i h$ . In order to answer this question, we would like to solve the problem [MC] with input  $\langle H, E, h, X^h \rangle$ , where  $X^h = \{h_1, h_5\}$  and  $h$  is just a parameter of the problem. This corresponds to solve the problem [EC] with input  $\langle V, X, E \rangle$ , where  $V = \{h_2, h_3, h_4\}$  and  $X = \{h_1, h_5\}$ . We thus look for some elements in  $V$  that could guarantee the existence of a strategy that includes  $h$  and none of the elements in  $X$ . We apply the system [CL]. For every vertex  $i \in V$  let us introduce the variable  $x_i \in \{0, 1\}$  which is equal to 1 if vertex  $i \in V$  is included in the clique. The vertex  $h_2$  is neither connected to  $h_3$  nor to  $h_4$ , therefore the constraints  $x_2 + x_3 \leq 1$  and  $x_2 + x_4 \leq 1$  are added. The only vertex not connected to  $h_1$  is vertex  $h_4$ , therefore it must be included in the clique:  $x_4 \geq 1$ . Both vertices  $h_2$  and  $h_3$  are connected to  $h_5$ , therefore at least one of them must be included in the clique:  $x_2 + x_3 \geq 1$ . The system [CL] for the graph of Fig. 9 is thus:

$$\begin{aligned}
 x_2 + x_3 &\leq 1, &> (h_2, h_3) \notin E \\
 x_2 + x_4 &\leq 1, &> (h_2, h_4) \notin E \\
 x_4 &\geq 1, &> (h_4, h_1) \notin E \\
 x_2 + x_3 &\geq 1, &> (h_2, h_5) \notin E, (h_3, h_5) \notin E \\
 x_2, x_3, x_4 &\in \{0, 1\}.
 \end{aligned}$$

**Fig. 9** Example. System [CL] is applied to the graph in figure, with  $V = \{h_2, h_3, h_4\}$  and  $X = \{h_1, h_5\}$

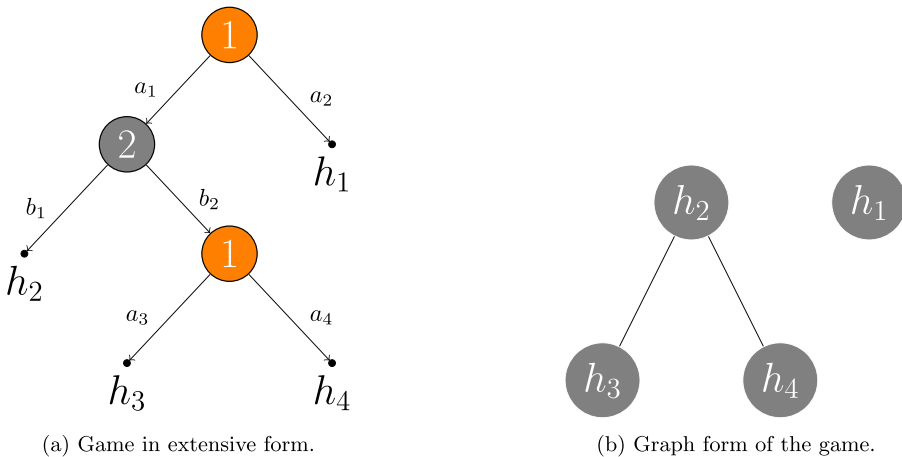


### A.5 Graph that Represents No Game

Let us consider the graph  $\langle H, E \rangle$  of Fig. 3 and show that it is not the graph form of an extensive-form game. If it were the graph form of an extensive-form game, any preferences could be assigned by the players to the outcomes, i.e. any utility function  $u$  can be chosen. We consider a utility function such that the preferences over the outcomes are respectively  $u_1 : h_A \succ_1 h_C \succ_1 h_D \succ_1 h_B$  and  $u_2 : h_B \succ_2 h_C \succ_2 h_A \succ_2 h_D$ . We show that if this graph were generated by a game, it would have no Nash equilibria. This is a contradiction, because every extensive-form game has at least one Nash equilibrium. The table of Fig. 3 shows the analysis for every outcome  $h \in \{h_A, h_B, h_C, h_D\}$ . Every line of the table shows that the composition of the vertex sets  $V$  and  $X$  for any pair of problems  $[EC]$  given  $\langle H, E \rangle$  and its complementary  $\langle H, E^C \rangle$  as input. For no outcome  $h$ , i.e. for no pair of problems at any given line, both problems has answer true. Indeed, neither  $h_B$  nor  $h_D$  can be the outcome of a Nash equilibrium, because they are the least favoured respectively by the first and by the second player, and there is no maximal clique on  $\langle H, E \rangle$  and on  $\langle H, E^C \rangle$  that consists of the singleton including them, i.e.  $\{h_B\}$  and  $\{h_D\}$  are not maximal cliques. In problem  $[EC]$  for  $h_D$  in graph  $\langle H, E \rangle$  we have that  $V = \emptyset$ , as well as for  $h_B$  in graph  $\langle H, E^C \rangle$ . Outcome  $h_A$  also is not a Nash equilibrium, because the only maximal clique on the graph which includes it is  $\{h_A, h_B\}$ , but  $h_B \succ_2 h_A$ . We observe that in problem  $[EC]$  defined for  $h_A$  in graph  $\langle H, E \rangle$  we have  $V = \emptyset$ . Finally, outcome  $h_C$  is not a Nash equilibrium, because the only maximal clique on the graph defined from  $\langle H, E^C \rangle$  is  $\{h_A, h_C\}$ , but  $h_A \succ_1 h_C$ . Analogously, in problem  $[EC]$  defined for  $h_C$  in graph  $\langle H, E^C \rangle$  we have  $V = \emptyset$ . Therefore there is no outcome corresponding to a Nash equilibrium, which is a contradiction. This graph represents no extensive-form game.

### A.6 [VS] does not Provide the Tightest Bound

Let us apply  $[VS]$  to the game of Fig. 10a. The sequences are respectively  $\Lambda_1 = \{\emptyset, a_1, a_2, a_1a_3, a_1a_4\}$  and  $\Lambda_2 = \{\emptyset, b_1, b_2\}$ . The optimal values are  $x_{a_1} = 1, x_{a_1a_3} = 1$ ,



**Fig. 10** Upper bound is not a Nash equilibrium. Preferences of the players over the outcomes are respectively:  $u_1 : h_3 \succ_1 h_4 \succ_1 h_2 \succ_1 h_1$  and  $u_2 : h_1 \succ_2 h_4 \succ_2 h_2 \succ_2 h_3$

$y_{b_2} = 1$  and  $u_1^{VS} = u_1(h_4)$ . Therefore the utility of the first player for outcome  $h_4$  provides an upper bound to the utility of the first player for any Nash equilibrium. However,  $h_4$  is not the realisation of a Nash equilibrium: since  $h_3 \succ_1 h_4$ , the first player would never choose action  $a_4$  over  $a_3$  in a Nash equilibrium. The reader can verify, by using Algorithm [EA] on the graph of Fig. 10b, that  $h_2$  is the only realisation of a Nash equilibrium.

## Appendix B: Proofs

### B.1 Proof of Theorem 7

(i) We prove first that for any Nash equilibrium the two maximal cliques in the statement do exist. If  $h$  is a realisation of a Nash equilibrium  $(\bar{s}_1, \bar{s}_2) \in S_1 \times S_2$  we have that for all  $s_1 \in S_1$  and  $s_2 \in S_2$ :

$$\begin{aligned} u_1(h) &= u_1(\bar{s}_1, \bar{s}_2) \geq u_1(s_1, \bar{s}_2), \\ u_2(h) &= u_2(\bar{s}_1, \bar{s}_2) \geq u_2(\bar{s}_1, s_2). \end{aligned}$$

We consider  $H(\bar{s}_1) \subset H$ . By Lemma 3,  $H(\bar{s}_1)$  forms a maximal clique in graph  $(H, E)$ . We thus define  $C_1 = H(\bar{s}_1)$ . Since  $(\bar{s}_1, \bar{s}_2) \mapsto h$ , we have that  $h \in C_1$ . For each  $h' \in C_1$  there is a strategy  $s_2 \in S_2$  such that  $(\bar{s}_1, s_2) \mapsto h'$ . Since  $h$  is a realisation of a Nash equilibrium, for each  $h' \in C_1$  we have that  $u_2(h) \geq u_2(h')$ . Therefore each  $h' \in C_1 \setminus X_2^h$  and, as a consequence, the vertex set  $C_1$  solves [MC] with input  $(H, E, h, X_2^h)$ . An analogous argument can be used to show that  $C_2 = H(\bar{s}_2)$  solves [MC] with input  $(H, E^C, h, X_1^h)$ .

(ii) We now prove that, if there are two vertex sets  $C_1 \subset H$  and  $C_2 \subset H$  that solve problem [MC] for, respectively,  $(H, E, h, X_2^h)$  and  $(H, E^C, h, X_1^h)$ , then  $h$  is a realisation of a Nash equilibrium. By Lemma 4, we have that there are  $\bar{s}_1 \in S_1$  and  $\bar{s}_2 \in S_2$  such that  $H(\bar{s}_1) = C_1$  and  $H(\bar{s}_2) = C_2$ . We recall that, since  $C_1 = H(\bar{s}_1)$ , for each  $s_2 \in S_2$  the strategy profile  $(\bar{s}_1, s_2) \mapsto h' \in C_1$ . Since  $C_1 \cap X_2^h = \emptyset$ , we have that  $h' \notin X_2^h$  and therefore  $u_2(h') \leq u_2(h)$ . Since  $\{h\} = H(\bar{s}_1) \cap H(\bar{s}_2)$ , the strategy profile  $(\bar{s}_1, \bar{s}_2)$  has realisation  $h$  satisfying:

$$u_2(\bar{s}_1, s_2) = u_2(h') \leq u_2(h) = u_2(\bar{s}_1, \bar{s}_2).$$

Analogously, we prove that for all  $s_1 \in S_1$ :

$$u_1(s_1, \bar{s}_2) \leq u_1(\bar{s}_1, \bar{s}_2).$$

Therefore  $(\bar{s}_1, \bar{s}_2)$  is a Nash equilibrium and  $h$  is its realisation.

### B.2 Proof of Proposition 9

(i) First, let us suppose that the problem [MC] has true as answer, i.e. there is a vertex set  $C \subset H \setminus X^h$  with  $h \in C$  that induces a maximal clique on  $(H, E)$ . By construction,  $C \subset V$  induces a clique on  $(V \cup X, E|_{V \cup X})$ . Moreover, the clique is maximal, i.e. there is no  $h'' \in H \setminus C$  such that for all  $h' \in C$  we have  $(h', h'') \in E$ . Therefore for all  $h'' \in X$  there is  $h' \in C$  such that  $(h', h'') \notin E$  and thus  $(h', h'') \notin E|_{V \cup X}$ .

(ii) Let us suppose now that the problem [EC] has true as answer, i.e. there is a vertex set  $C' \subset V$  that induces a clique on  $(V \cup X, E|_{V \cup X})$  such that for all  $h'' \in X$  there is  $h' \in C'$  such that  $(h', h'') \notin E|_{V \cup X}$ . Let us consider  $C'' = C' \cup \{h\}$ , which induces a clique on graph  $(H, E)$ . For all  $h'' \in X^h$  either (a)  $h'' \in X$  and thus there is an element  $h' \in C'$  such that  $(h', h'') \notin E$  or (b)  $h'' \in X^h \setminus X$  and thus  $(h, h'') \notin E$ . Therefore for all  $h'' \in X^h$  there is

$h' \in C'' = C' \cup \{h\}$  such that  $(h', h'') \notin E$ . If  $C''$  induces not only a clique, but a maximal one on  $\langle H, E \rangle$  we have the proof. If  $C''$  induces a non-maximal clique, there is a vertex set  $C'''$  such that  $C'' \subset C'''$  that induces a maximal clique on  $\langle H, E \rangle$ . By construction  $C''' \cap X = \emptyset$ , which completes the proof.

### B.3 Best and Worst Nash Equilibrium

---

#### Algorithm 3 [BNE] (WNE) best (worst) Nash equilibrium

---

```

INPUT: Game in graph form  $\Gamma = \langle H, E, u \rangle$ 
 $H^{\geq 1} = \text{order}(H, \geq 1) \triangleright$  Elements  $h \in H$  are ordered from best to worst [from worst to best] w.r.t. player
1:  $h \geq_1 h'$  iff  $u_1(h) \geq u_1(h')$  [iff  $u_1(h) \leq u_1(h')$ ]
index = 0
do
   $h = H^{\geq 1}(\text{index})$   $\triangleright$  Consider outcome  $h$ 
   $X_1 = \{h' \in H \mid (h', h) \notin E, u_1(h') > u_1(h)\}$   $\triangleright$  Init vertex sets and excluding sets
   $X_2 = \{h'' \in H \mid (h'', h) \in E, u_2(h'') > u_2(h)\}$ 
   $V_1 = \{h' \in H \setminus X_2 \mid (h', h) \in E\}$ 
   $V_2 = \{h'' \in H \setminus X_1 \mid (h'', h) \notin E\}$ 
  solve [NE] giving  $\langle V_1, V_2, X_1, X_2, E \rangle$  as input
  index = index + 1
while [NE] has no solution
OUTPUT:  $h$ 

```

---

**Author Contributions** All the authors contributed equally to this work.

**Funding** Not applicable.

**Data Availability** The dataset used in the numerical experiments is available online (cf. [31]).

### Declarations

**Conflict of interest** The authors have no relevant financial or non-financial interests to disclose.

**Ethical Approval** Not applicable.

### References

1. Kuhn HW, Tucker AW (1953) Contributions to the theory of games. Princeton University Press, Princeton
2. Long NV (2011) Dynamic games in the economics of natural resources: a survey. *Dyn Games Appl* 1:115–148
3. Baird DG, Gertner RH, Picker RC (1998) Game theory and the law. Harvard University Press, Cambridge
4. Etesami SR, Başar T (2019) Dynamic games in cyber-physical security: an overview. *Dyn Games Appl* 9(4):884–913
5. Nash JF Jr (1950) Equilibrium points in n-person games. *Proc Natl Acad Sci* 36(1):48–49
6. Selten R (1965) Spieltheoretische behandlung eines oligopolmodells mit nachfragerträgeheit: teil i: bestimmung des dynamischen preisgleichgewichts. *Z Gesamte Staatswiss J Inst Theor Econ H* 2:301–324
7. Selten R (1978) The chain store paradox. *Theory Decis* 9(2):127–159
8. Pettit P, Sugden R (1989) The backward induction paradox. *J Philos* 86(4):169–182
9. Binmore K, McCarthy J, Ponti G, Samuelson L, Shaked A (2002) A backward induction experiment. *J Econ Theory* 104(1):48–88

10. Daskalakis C, Goldberg PW, Papadimitriou CH (2009) The complexity of computing a Nash equilibrium. *SIAM J Comput* 39(1):195–259
11. Papadimitriou CH (1992) On inefficient proofs of existence and complexity classes. *Ann Discret Math* 51:245–250. [https://doi.org/10.1016/S0167-5060\(08\)70637-X](https://doi.org/10.1016/S0167-5060(08)70637-X)
12. Lemke CE, Howson JT Jr (1964) Equilibrium points of bimatrix games. *J Soc Ind Appl Math* 12(2):413–423
13. Avis D, Fukuda K (1992) A pivoting algorithm for convex hulls and vertex enumeration of arrangements and polyhedra. *Discret Comput Geom* 8(3):295–313
14. Avis D, Rosenberg GD, Savani R, Von Stengel B (2010) Enumeration of Nash equilibria for two-player games. *Econ Theory* 42(1):9–37
15. Widger J, Grosu D (2009) Parallel computation of Nash equilibria in n-player games. In: 2009 international conference on computational science and engineering, vol 1. IEEE, pp 209–215
16. Von Stengel B (1996) Efficient computation of behavior strategies. *Games Econ Behav* 14(2):220–246
17. Koller D, Megiddo N, Von Stengel B (1996) Efficient computation of equilibria for extensive two-person games. *Games Econ Behav* 14(2):247–259
18. Audet C, Belhaiza S, Hansen P (2009) A new sequence form approach for the enumeration and refinement of all extreme Nash equilibria for extensive form games. *Int Game Theory Rev* 11(04):437–451
19. Govindan S, Wilson R (2002) Structure theorems for game trees. *Proc Natl Acad Sci* 99(13):9077–9080
20. Herings PJ-J, Peeters R (2010) Homotopy methods to compute equilibria in game theory. *Econ Theory* 42(1):119–156
21. Argenziano R, Schmidt-Dengler P (2014) Clustering in-player preemption games. *J Eur Econ Assoc* 12(2):368–396
22. Selten R, Mitzkewitz M, Uhlich GR (1997) Duopoly strategies programmed by experienced players. *Econom J Econom Soc* 65:517–555
23. Hassin R, Yovel U (2015) Sequential scheduling on identical machines. *Oper Res Lett* 43(5):530–533
24. Fudenberg D, Tirole J (1991) *Game theory*. The MIT Press, Cambridge
25. Kreps DM, Wilson R (1982) Sequential equilibria. *Econom J Econom Soc* 23:863–894
26. Fudenberg D, Levine D (1983) Subgame-perfect equilibria of finite-and infinite-horizon games. *J Econ Theory* 31(2):251–268
27. Koller D, Megiddo N (1992) The complexity of two-person zero-sum games in extensive form. *Games Econ Behav* 4(4):528–552
28. Zappalà P, Benhamiche A, Chardy M, De Pellegrini F, Figueiredo R (2022) A timing game approach for the roll-out of new mobile technologies. In: 2022 20th international symposium on modeling and optimization in mobile, Ad hoc, and wireless networks (WiOpt), pp 217–224. <https://doi.org/10.23919/WiOpt56218.2022.9930538>
29. Kratsch D, Liedloff M (2007) An exact algorithm for the minimum dominating clique problem. *Theor Comput Sci* 385(1–3):226–240
30. Makhorin A (2008) GLPK (GNU linear programming kit). <http://www.gnu.org/s/glpk/glpk.html>
31. Zappalà P (2023) GT library 1.0. Dataset of extensive-form games. <https://doi.org/10.5281/zenodo.7628367>. <https://github.com/paolozapp/gtlibrary>

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.