



HAL
open science

Computing Nash equilibria in extensive-form games with more than two players

Paolo Zappalà, Amal Benhamiche, Matthieu Chardy, Francesco De Pellegrini,
Rosa Figueiredo

► **To cite this version:**

Paolo Zappalà, Amal Benhamiche, Matthieu Chardy, Francesco De Pellegrini, Rosa Figueiredo. Computing Nash equilibria in extensive-form games with more than two players. 2023. hal-03962754

HAL Id: hal-03962754

<https://hal.science/hal-03962754>

Preprint submitted on 1 Mar 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Computing Nash equilibria in extensive-form games with more than two players

PAOLO ZAPPALÀ, AMAL BENHAMICHE, MATTHIEU CHARDY, FRANCESCO DE PELLEGRINI, ROSA FIGUEIREDO

Identifying efficiently the Nash equilibria of extensive-form games with perfect recall and perfect information is a known open problem. In this paper, we introduce a new representation of a N -player extensive-form game as an undirected complete graph of its outcomes. The Nash equilibria can hence be characterized directly on the graph of the outcomes, without naming the corresponding strategies. Such approach lets us introduce two distinct algorithms for the enumeration of Nash equilibria. The first algorithm iterates over the outcomes and determines on a subgraph whether each of them is a realisation of a Nash equilibrium. The second one generalises the backward induction algorithm and recursively enumerates all the outcomes of the Nash equilibria of the game. The recursive algorithm is more efficient, but it requires the exploration of the entire game tree, while the iterative algorithm can be used to provide results on any subset of outcomes.

1 INTRODUCTION

In extensive-form games with perfect recall and perfect information a finite set of agents, called *players*, observe in turns each other's actions and pick one of the possible subsequent actions available to them [Kuhn and Tucker, 1953]. A *strategy* of a player consists of picking one action whenever they ought to during the game. A Nash equilibrium is a combination of strategies such that no player has an incentive to change her own strategy. Every game admits at least one Nash equilibrium [Nash Jr, 1950], but no results have been given on the identification of all Nash equilibria of a generic extensive-form game. Algorithms for the enumeration of Nash equilibria are provided for two-player extensive-form games [Zappalà et al., 2023]. Extensive-form games always admit a specific Nash equilibrium, called *subgame perfect equilibrium* [Kuhn and Tucker, 1953]. Such equilibrium can be computed by the algorithm called *backward induction* [Cayley, 1875], but it does not give insights on other Nash equilibria. In this paper we introduce a representation of extensive-form games as an undirected complete graph of their outcomes. We provide a necessary and sufficient condition for an outcome to be a realisation of a Nash equilibrium. Moreover, we design an extension of the backward induction algorithm which can be proved to provide the complete enumeration of all Nash Equilibria of a generic extensive-form game. Section 2 introduces the reader to extensive-form games and all the formal definitions for the aforementioned objects. Section 3 discusses the new graph representation of extensive-form games. Section 4 presents an algorithm to determine if an outcome is a realisation of a Nash equilibrium. In Section 5 we introduce an algorithm to enumerate the Nash equilibria, which is proved to be polynomial in the size of the game. Finally, Section 6 ends the paper with possible directions of future research.

2 EXTENSIVE-FORM GAMES

In this section we provide the definitions for extensive-form games with perfect recall and perfect information as in [Zappalà et al., 2023]. An extensive-form game develops in a finite number of stages. At every stage there exists a designated player $i \in \mathcal{I}$ who observes a *history* h' , i.e., the sequence of actions occurring up to the stage. The designated player has available a set of actions $\mathcal{A}(h')$ at this stage. We denote by $P(h')$ the designated player observing the history h' . When the set of available actions is empty ($\mathcal{A}(h) = \emptyset$) the game ends, i.e., the sequence of actions leading to this stage h corresponds to an *outcome* of the game. We call H' the set of histories and $H \subset H'$ the set of the outcomes. Every outcome is evaluated by every player $i \in \mathcal{I}$ through a function, called *utility function*, $u_i : H \rightarrow \mathbb{R}$. We also write $h_A >_i h_B$ for $h_A, h_B \in H$ when $u_i(h_A) > u_i(h_B)$, or $h_A \sim_i h_B$ when $u_i(h_A) = u_i(h_B)$. Following [Zappalà et al., 2023], we denote by $h' + h'' = (a_1, a_2, \dots, a_{m'}, b_1, b_2, \dots, b_{m''})$ the concatenation of two vectors of actions $h' = (a_1, a_2, \dots, a_{m'})$ and $h'' = (b_1, b_2, \dots, b_{m''})$. We also denote by $h''' = h' \cap h''$ the lowest common prefix of the two vectors of actions h' and h'' , shortly referred as *prefix* in the following. The full definition of extensive-form game is therefore:

Definition 2.1 (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 A , i.e. for all $a \in A$, 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 $u_i : H \rightarrow \mathbb{R}$ the utility function of player $i \in \mathcal{I}$.

Since in the literature the representation of the game is the game-tree of possible histories, it is customary to call a *node* a history observed by a player. Analogously, we alternatively call an outcome the final node (or *leaf*) and the vector of actions leading to it. When a node is not final, the rest of the game starting at it is itself a game called *subgame*. Indeed, once a player has observed a history $h^k \in H'$ with $k \in \{1, \dots, |H'|\}$ at a given stage, we know that the alternative actions up to this stage have not been played. We thus consider the possible histories that derive from h^k .

Example. Let us consider the game-tree of Fig. 1. If player 3 observes the player 1 choosing $h^k = (L)$, we obtain the subgame framed in Fig. 1. Definition 2.2 formalises the concept of subgame.

Definition 2.2 (subgame). Given an extensive-form game $\Gamma = \langle \mathcal{I}, \mathcal{A}, H', H, P, u \rangle$ and a history h^k we call subgame the game $\Gamma(h^k) = \langle \mathcal{I}^k, \mathcal{A}^k, H'^k, H^k, P^k, u^k \rangle$ where:

- $\mathcal{I}^k = \mathcal{I}$, i.e., the set of players is the same as in the original game;
- $H'^k = \{h'^k, \exists h' \in H', h' = h^k + h'^k\}$, the set of histories share the same prefix h^k ;
- $\mathcal{A}^k(h'^k) = \mathcal{A}(h^k + h'^k)$;
- $H^k = \{h \in H'^k \mid \mathcal{A}^k(h) = \emptyset\}$;
- $P^k(h'^k) = P(h^k + h'^k)$;
- $u^k = (u_i^k)_{i \in \mathcal{I}}$, with $u_i^k(h) = (h^k + h)$.

As in [Zappalà et al., 2023] we only consider pure strategies, from now simply called *strategies*, as they are enough to compute all equilibria (cf. Theorem 1 of [Audet et al., 2009]). A strategy is a function that maps every history observed by a player to an action.

Definition 2.3 (strategy). Given a game $\Gamma = \langle \mathcal{I}, \mathcal{A}, H', H, P, u \rangle$ and a player $i \in \mathcal{I}$, let $H'_{P=i} = \{h' \in H' \mid H|P(h) = i\}$ be the histories at which the player i acts. A strategy $s_i \in S_i$ is a function $s_i : h' \in H'_{P=i} \mapsto a \in \mathcal{A}(h')$ that maps every observed history $h' \in H'_{P=i}$ to one of the actions $a \in \mathcal{A}(h')$ available to the player.

We call *strategy profile* a N -tuple of strategies $s = \langle s_1, s_2, \dots, s_N \rangle$, one for each player. We denote by $S = S_1 \times S_2 \times \dots \times S_N$ the set of all strategy profiles. If every player chooses a strategy, one single action is picked at every history; therefore, given a strategy profile, the actions chosen by the players lead to a single outcome. We denote by $s \mapsto h$ the outcome $h \in H$ of a strategy profile $s \in S$. When a player picks a strategy, she limits the set of possible outcomes. We define such specific set in Definition 2.4.

Definition 2.4 (outcomes of a strategy). Given a game $\Gamma = \langle \mathcal{I}, \mathcal{A}, H', H, P, u \rangle$ and a strategy $s_i \in S_i$ of a player $i \in \mathcal{I}$, the set of outcomes $H(s_i) \subset H$ of strategy s_i is

$$H(s_i) = \{h \in H \mid \exists s' \in S, s'_i = s_i, s' \mapsto h\}.$$

We also write $H(\langle s_j \rangle_{j \in J}) = \cap_{j \in J} H(s_j)$ to indicate the possible outcomes of a vector of strategies $\langle s_j \rangle_{j \in J}$ for some players $J \subset \mathcal{I}$. We write $\langle s_j \rangle_{j \in J} = s_{-i}$ when $J = \mathcal{I} \setminus \{i\}$. Clearly for a strategy profile $s \in S$ the set $H(s) = \{h\}$ is a singleton. Furthermore, with some abuse of notation, let $u_i(s) := u_i(s \mapsto h)$ denote the utility of player i under a certain strategy profile s . A strategy profile is a Nash equilibrium if no player can increase her utility by changing unilaterally her strategy.

Definition 2.5 (Nash equilibrium). Given a game $\Gamma = \langle \mathcal{I}, \mathcal{A}, H', H, P, u \rangle$, a strategy profile $\{\bar{s}_i\}_{i \in \mathcal{I}}$ is a Nash equilibrium if for every $i \in \mathcal{I}$ and for all $s_i \in S_i$ it holds $u_i(\bar{s}_i, \bar{s}_{-i}) \geq u_i(s_i, \bar{s}_{-i})$.

The recursive algorithm proposed in this work is inspired by the backward induction (BI) algorithm (cf. Algorithm 1), which is the most known algorithm to compute a Nash equilibrium in extensive-form games [Cayley, 1875]. However, the backward induction algorithm provides only a specific subset of Nash equilibria, i.e., the *subgame perfect equilibria* (SPE). A subgame perfect

ALGORITHM 1: Backward induction (BI)**Input:** A game $\Gamma = \langle \mathcal{I}, \mathcal{A}, H', H, P, u \rangle$ and its root $h^0 = \emptyset \in H'$.**Output:** The set of subgame perfect equilibria SPE .**if** $|H| = 1$ **then** $SPE = H$;**else** $i = P(h^0)$;

// The player acting at the vertex

 $\langle \Gamma^k, SPE^k, h'_k \rangle_{a_k \in \mathcal{A}(h^0)} \leftarrow \emptyset$;**for** $a_k \in \mathcal{A}(h^0)$ **do** $\Gamma^k = \Gamma(h^0 + (a_k))$;// The subgame that follows actions a_k $SPE^k = BI(\Gamma^k)$; $h'_k \in \arg \min_{h' \in SPE^k} u_i(h')$; // The lowest utility u_i a SPE can achieve in Γ^k **end** $SPE = \{h \in \cup SPE^k \mid \forall k, h \succeq_i h'_k\}$; // Outcome h is preferred by player i to any other SPE**end**

equilibrium is a Nash equilibrium for every subgame [Selten, 1965]. Every extensive-form game with perfect recall and perfect information admits a subgame perfect equilibrium in pure strategies. The backward induction selects, starting from the leaves of the game-tree, the outcomes that are most favourite by the player acting at a given node. The subgame perfect equilibrium might not be unique, so the candidate outcome h must be preferred among at least one possible selection of candidate outcomes $\{h'_k \in SPE^k\}$ of the other subgames. The value of the corresponding outcomes thus propagates upwards towards the root of the game-tree as exemplified next.

Example. Let us consider the game Γ represented by tree of Fig. 1a. The preferences of the players w.r.t. the outcomes are indicated in the caption. Let us compute the subgame perfect equilibria of the game by applying the BI algorithm. The algorithm starts from the leaves of the tree. Player 1 prefers h_2 to h_1 ($h_2 \succ_1 h_1$), player 2 has no strict preference between h_3 and h_4 ($h_3 \sim_2 h_4$) and player 3 prefers h_8 to h_7 ($h_8 \succ_3 h_7$). The outcomes h_2 , $\{h_3, h_4\}$ and h_8 are the SPE of the respective subgames, as shown in Fig. 1b. At the second stage of the tree, players 3 and 2 prefer respectively h_2 to h_3 and h_4 ($h_2 \succ_3 h_3 \sim_3 h_4$), h_5 to h_6 and h_8 ($h_5 \succ_2 h_8 \succ_2 h_6$). Finally, at the root of the tree, player 1 prefers h_2 to h_5 ($h_2 \succ_1 h_5$). The (here unique) subgame perfect equilibrium of the game is h_2 .

In the following section we focus on identifying the outcomes of the Nash equilibria. Let us consider a strategy profile $s \in S$. We recall that its realisation is the only element $h \in H$ belonging to the set of its possible outcomes $H(s)$. For any strategy profile originated by a unilateral deviation $s' \in S$ it must hold $s'_i \neq s_i$ and $s'_{-i} = s_{-i}$ for one and only one $i \in \mathcal{I}$. Therefore the realisation of any possible unilateral deviation, namely $h' \in H$, belongs to the set of possible outcomes $H(s_{-i})$ of the strategies $s_{-i} \in S_{-i} = \times_{j \in \mathcal{I} \setminus \{i\}} S_j$ of all players but the one deviating $i \in \mathcal{I}$. We thus characterise the realisation of a Nash equilibrium over the sets of outcomes.

LEMMA 2.6. *Given a game $\Gamma = \langle \mathcal{I}, \mathcal{A}, H', H, P, u \rangle$, an outcome $h \in H$ is a realisation of a Nash equilibrium if and only if there exists a strategy profile $s \in S$ such that $H(s) = \{h\}$ and for each $i \in \mathcal{I}$ and $h' \in H(s_{-i})$ it holds $u_i(h) \geq u_i(h')$.*

PROOF. The direct implication is obvious. For the converse let us consider the strategy profile $s \in S$ whose existence is assumed in the statement. Observe that set $H(s_{-i}) \setminus \{h\} = \{h' \in H \mid \exists s'_i \in S_i, s = \langle s_i, s_{-i} \rangle \mapsto h', h' \neq h\}$ is the set of the outcomes of the strategy profiles of type s' where $s' = \langle s'_i, s_{-i} \rangle$, $s'_i \neq s_i$, i.e., of strategy profiles which are unilateral deviations from s . Hence, for any such strategy profile it holds $u_i(s'_i, s_{-i}) = u_i(h') \leq u_i(h) = u_i(s_i, s_{-i})$. \square

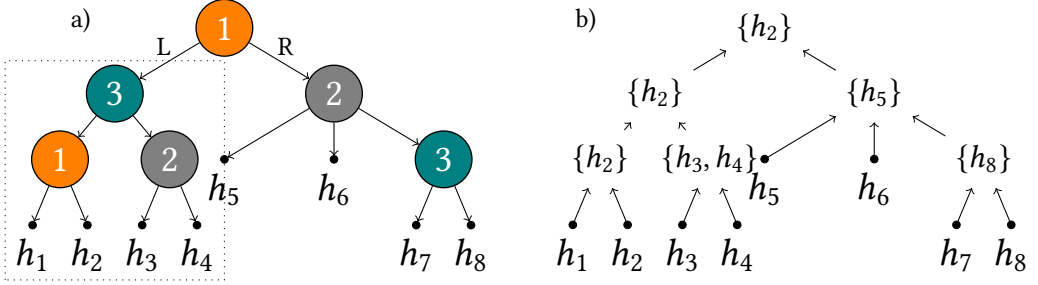


Fig. 1. **Example.** a) 3-player game in extensive form. Preferences of the players over the outcomes are respectively: $u_1 : h_6 \succ_1 h_7 \succ_1 h_8 \succ_1 h_3 \succ_1 h_4 \succ_1 h_2 \succ_1 h_1 \succ_1 h_5$, $u_2 : h_5 \succ_2 h_8 \succ_2 h_7 \succ_2 h_6 \succ_2 h_2 \succ_2 h_3 \sim_2 h_4 \succ_2 h_1$ and $u_3 : h_8 \succ_3 h_7 \succ_3 h_6 \succ_3 h_2 \succ_3 h_5 \succ_3 h_3 \succ_3 h_1 \succ_3 h_4$. A subgame is framed. b) Application of the backward induction to the game. The subgame perfect equilibrium of the game is h_2 .

3 GRAPH FORM

In this section we characterize the set of outcomes $H(s_i)$ of a strategy $s_i \in S_i$ of a player $i \in I$. To this aim we introduce a new representation of extensive-form games as an undirected graph of their outcomes. Let us consider the game of Fig. 1. We observe that at the root the first player chooses between going left (L) and going right (R). Thus, the first player cannot choose a pure strategy that can lead to both h_1 and h_8 : since two different actions must be chosen, two different strategies must be taken. Analogously, the second player cannot choose a strategy whose outcomes include both h_3 and h_4 . With this argument we can conclude that two outcomes cannot be included in the set of outcomes of the same strategy of a player $i \in I$ if the paths from the root to such outcomes are separated at a node where player i acts. Given two outcomes $h, h' \in H$, the node at which the paths from the root to them are separated is the history corresponding to the common prefix $h \cap h'$, and the player acting at such node is $P(h \cap h')$. Moreover, there is always a way for the other players to let player 1 choose between h_1 and h_8 , i.e., there is a combination of strategies $s_{-1} \in S_{-1} = S_2 \times S_3$ such that $h_1, h_8 \in H(s_{-1})$. We can formalise the above arguments in the following theorem.

THEOREM 3.1. *Given a game $\Gamma = \langle I, \mathcal{A}, H', H, P, u \rangle$ and two outcomes $h, h' \in H$, the following three propositions are equivalent:*

- (1) $i = P(h \cap h')$;
- (2) There is no $s_i \in S_i$ such that $h, h' \in H(s_i)$;
- (3) There exists a set of strategies $s_{-i} \in S_{-i}$ such that $h, h' \in H(s_{-i})$.

PROOF. Let us represent outcomes h and h' as the two sequences of actions leading to them, i.e., $h = (a_k)_{k \in \{1, \dots, K\}}$ and $h' = (a'_k)_{k \in \{1, \dots, K\}}$, respectively. We denote h^r and h'^r the histories which are the prefix of size r of h and h' , respectively. By definition the prefix $h \cap h'$, of size \bar{r} , is the history such that $h^r = h'^r$ for $r \leq \bar{r}$ and $h^{\bar{r}+1} \neq h'^{\bar{r}+1}$.

(1) \Rightarrow (3). Let us consider strategy profile $s = \langle s_j \in S_j \rangle_{j \in I}$ defined as follows: $s_j(h^r) = a_{r+1}$ and $s_j(h'^r) = a'_{r+1}$ for all $r \neq \bar{r}$ such that respectively $P(h^r) = j$ or $P(h'^r) = j$. For all other nodes of the game tree, the actions are chosen at random. Let us consider two strategies $s_i, s'_i \in S_i$ of player $i = P(h \cap h')$ and such that $s_i(a_{\bar{r}}) = a_{\bar{r}+1}$ and $s'_i(a'_{\bar{r}}) = a'_{\bar{r}+1}$. By construction $\langle s_1, \dots, s_i, \dots, s_N \rangle \mapsto h$ and $\langle s_1, \dots, s'_i, \dots, s_N \rangle \mapsto h'$ and thus for all $j \neq i$ it holds $h, h' \in H(s_j)$.

(2) \Rightarrow (1). We prove it by contradiction. If $P(h \cap h') = j \neq i$ we would have that, since (1) \Rightarrow (3), there exists $s_i \in S_i$ such that $h, h' \in H(s_i)$, against the assumptions in (2).

(3) \Rightarrow (2). If there is a strategy $s_i \in S_i$ such that $h, h' \in H(s_i)$, then we have $\{h, h'\} \subseteq H(s)$, i.e., strategy profile $\langle s_i, s_{-i} \rangle$ would have more than one outcome, which is absurd. \square

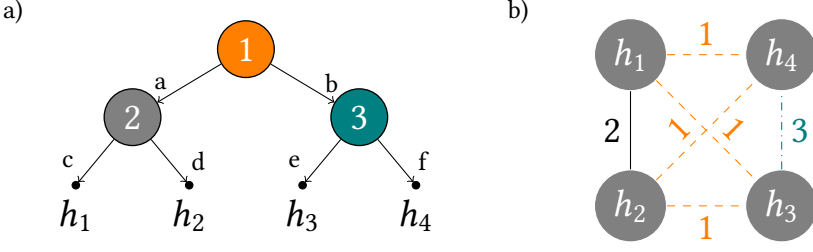


Fig. 2. **Example of graph form.** a) Game in extensive form. b) Game in graph form.

The expression $P(h \cap h')$ is capital for the following part of the analysis. Let us highlight it by defining function

$$I : H \times H \rightarrow \mathcal{I} \text{ where } I(h, h') = P(h \cap h').$$

It maps the pair of outcomes $h, h' \in H$ with $h \neq h'$ to the player $i = I(h, h')$ that separates their paths from the root of the game tree. In order to define the set of possible outcomes $H(s_i)$ of a strategy $s_i \in S_i$ of a player $i \in \mathcal{I}$, it is possible to use directly function I to select which elements $h \in H$ can or cannot belong to $H(s_i)$. We therefore represent the game as a complete graph whose vertex set is made by the game outcomes H and each edge $(h, h') \in H^2$ is labeled with the label of player $I(h, h')$.

Definition 3.2. Given an extensive-form game $\Gamma = \langle \mathcal{I}, \mathcal{A}, H', H, P, u \rangle$ the graph form $\langle H, I, u \rangle$ is described by the complete edge-labeled graph with vertex set H , where every edge $(h, h') \in H^2$ with $h \neq h'$ has label $I(h, h') = P(h \cap h') \in \mathcal{I}$ and the utility function $u : H \rightarrow \mathbb{R}^N$.

Example. Let us observe the game of Fig. 2 with its graph form. In this game each player acts at only one node, therefore we represent with s^a the strategy that chooses action a . The paths from the root to outcomes h_1 and h_3 are split by player 1, who can choose whether to go left (strategy s^a) or to go right (strategy s^b). We thus write $I(h_1, h_3) = 1$ and we assign the label 1 to arc (h_1, h_3) . Analogously, the paths belonging to outcomes h_3 and h_4 are split by player 3 who can go either left (strategy s^e) or right (strategy s^f). Therefore we write $I(h_3, h_4) = 3$, assigning label 3 to arc (h_3, h_4) .

As anticipated, we would like to define the set of outcomes of a strategy by using the function $I : H \times H \rightarrow \mathcal{I}$. The graph form contains all the possible values of function $I(h, h')$ for every couple of outcomes $(h, h') \in H \times H$ of the game. Let us analyse our example and then conclude how to characterize the set of outcomes of a strategy directly on the graph of the game.

Example. For the game of Fig. 2a let us enlist all the strategies, their possible outcomes and then observe the corresponding labelling on the graph. Player 1 has two strategies: s^a and s^b . If player 1 picks strategy s^a the only possible outcomes are $H(s^a) = \{h_1, h_2\}$, while if she picks strategy s^b we have $H(s^b) = \{h_3, h_4\}$. Player 2 has two strategies: s^c and s^d . If player 2 chooses strategy s^c , she limits the possible outcomes to $H(s^c) = \{h_1, h_3, h_4\}$, while if she chooses s^d we have $H(s^d) = \{h_2, h_3, h_4\}$. Finally for player 3 we have that $H(s^e) = \{h_1, h_2, h_3\}$ and $H(s^f) = \{h_1, h_2, h_4\}$. Let us consider the graph form depicted in Fig. 2b and remove all arcs with the same label $i \in \mathcal{I}$. Formally, let us define the graph $\langle H, E|_{\neq i} \rangle$ for each player $i \in \mathcal{I}$, with $E|_{\neq i} = \{(h, h') \in H^2 : I(h, h') \neq i\}$ excluding all the arcs $(h, h') \in E$ such that $I(h, h') = i$. Let us consider $i = 1$ and observe the strategies $s_1 = s^a$ and $s_1 = s^b$. Their outcomes $H(s^a) = \{h_1, h_2\}$ and $H(s^b) = \{h_3, h_4\}$ form cliques over $\langle H, E|_{\neq 1} \rangle$. This is compliant with Theorem 3.1, for which given a strategy s_i for all elements $h, h' \in H(s_i)$ we have that $I(h, h') \neq i$ and thus $(h, h') \in E|_{\neq i}$. Moreover, as proved next, with an argument similar to the one introduced in [Zappalà et al., 2023] we conclude that such cliques are maximal.

The above argument can be made stronger: not only the outcomes of a strategy $s_i \in S_i$ form a maximal clique over $\langle H, E|_{\neq i} \rangle$, but also for every set $C \subset H$ inducing a maximal clique on $\langle H, E|_{\neq i} \rangle$ there is a corresponding strategy $s_i \in S_i$ with such set of outcomes $C = H(s_i)$.

LEMMA 3.3. *Given an extensive-form game $\Gamma = \langle \mathcal{I}, \mathcal{A}, H', H, P, u \rangle$ and its graph form $\langle H, I, u \rangle$, let us consider $\langle H, E|_{\neq i} \rangle$ the graph with H as vertices and $E|_{\neq i} = \{(h, h') \in H^2, I(h, h') \neq i\}$ for some player $i \in \mathcal{I}$. There is a bijection between the set of maximal cliques of $\langle H, E|_{\neq i} \rangle$ and the set of outcomes of the player's strategies $H(S_i) = \{H(s_i) | s_i \in S_i\}$.*

PROOF. Let $C \subset H$ induce a maximal clique on graph $\langle H, E|_{\neq i} \rangle$. Let us define a strategy $s_i \in S_i$ such that $C \subset H(s_i)$. We use the same constructive argument as done in Theorem 3.1: for all $h \in C$, and for all k -prefix h^k such that $P(h^k) = i$, we can fix $s_i(h^k) = a_k$ such that $h^k + (a_k) = h^{k+1}$; indeed, since C is a clique, for every other $h' \in C$ such that h^k is a prefix of h' we have that h^{k+1} is a prefix also for h' . For all remaining nodes of the tree, actions can be fixed at random. Let $h'' \in H \setminus C$: by construction there is $h \in C$ such that $I(h, h') = i$, so that $h'' \notin H(s_i)$ and thus $H(s_i) = C$.

We now prove the opposite, i.e., the outcomes of a strategy of player $i \in \mathcal{I}$ define a maximal clique over $\langle H, E|_{\neq i} \rangle$. Consider a strategy $s_i \in S_i$ and the set of its outcomes $H(s_i)$. We apply Theorem 3.1: for every $h, h' \in H(s_i)$ it holds $I(h, h') \neq i$. By definition, all the elements $h, h' \in H(s_i)$ are connected in $\langle H, E|_{\neq i} \rangle$, i.e., $H(s_i)$ forms a clique. If this clique is not maximal, there exists $C \subset H$ with $H(s_i) \subsetneq C$ that induces a maximal clique over $\langle H, E|_{\neq i} \rangle$. As done before, we define $s'_i \in S_i$ a strategy such that $H(s'_i) = C$. We have thus $H(s_i) \subsetneq H(s'_i)$. This is absurd. Indeed, since $s_i \neq s'_i$, there exists $h^k \in H(s'_i)$ with $P(h^k) = i$ such that $s_i(h^k) \neq s'_i(h^k)$. Let us consider an element $h \in H(s_i)$ such that $h^k + s_i(h^k)$ is a prefix of h : such element exists because the subgame Γ^k starting from node $h^k + s_i(h^k)$ must be not empty. We have that $h \in H(s_i)$ and $h \notin H(s'_i)$, hence proving the contradiction. \square

The above result characterises the players' strategies in terms of a structural property of the graph form of the game. Let us recall that Lemma 2.6 defines a realisation of the Nash equilibria over the set of outcomes of the strategies, which are then characterised on the graph in Lemma 3.3. We combine the two lemmas to discuss over the graph whether a candidate outcome $h \in H$ is the realisation of a Nash equilibrium $s \in S$.

THEOREM 3.4. *Given a game in its graph form $\langle H, I, u \rangle$, let us consider $\langle H, E|_{\neq i} \rangle$ defined for each player $i \in \mathcal{I}$. An outcome $h \in H$ is a realisation of a Nash equilibrium if and only if there are sets $\{C_i \subset H\}_{i \in \mathcal{I}}$ that induce maximal cliques respectively over the graphs $\{\langle H, E|_{\neq i} \rangle\}_{i \in \mathcal{I}}$ such that:*

- i. $h \in \bigcap_{i \in \mathcal{I}} C_i$;
- ii. $\forall h' \in H \setminus \{h\}$ and $i = I(h, h')$ at least one of the two conditions holds: a) $u_i(h) \geq u_i(h')$ or b) $h' \notin \bigcap_{j \in \mathcal{I} \setminus \{i\}} C_j$.

PROOF. Let us consider a strategy $s \in S$ as for Lemma 2.6. First, for all $i \in \mathcal{I}$ it must hold $h \in H(s_i)$. Second, for any other outcome $h' \neq h$, if $I(h, h') = i$ we have that $h \in H(s_i)$ implies that $h' \notin H(s_i)$ (cf. Theorem 3.1). Therefore the only condition that allows $h' \in H(s_{-i}) = \bigcap_{j \in \mathcal{I} \setminus \{i\}} H(s_j)$ is that $I(h, h') = i$. Any outcome $h' \neq h$ must thus fulfill at least one of the two conditions: a) $u_i(h) \geq u_i(h')$ no matter if $h \in \bigcap_j H(s_j)$ or not, or b) $h' \notin \bigcap_j H(s_j)$. Finally, from Lemma 3.3 we know that the existence of every set $H(s_i)$ depends on the existence of a set $C_i = H(s_i)$ that forms a maximal clique on graph $\langle H, E|_{\neq i} \rangle$. \square

Example. Let us apply Theorem 3.4 to the game-graph of Fig. 2b for outcome $h_2 \in H$. Three cliques have to be identified C_1, C_2 and C_3 . The only set inducing a maximal clique on $\langle H, E|_{\neq 1} \rangle$ such that h_2 belongs to it is $C_1 = \{h_1, h_2\}$. Analogously, the only maximal clique on $\langle H, E|_{\neq 2} \rangle$

including h_2 is $C_2 = \{h_2, h_3, h_4\}$. Finally, the maximal cliques on $\langle H, E|_{\neq 3} \rangle$ are $C_3 = \{h_1, h_2, h_3\}$ and $C_3 = \{h_1, h_2, h_4\}$. Therefore $h_1 \in C_1 \cap C_3$ and the condition $u_2(h_2) \geq u_2(h_1)$ is necessary. On the other hand, either $h_3 \in C_2 \cap C_3$ and $h_4 \in C_2$ or $h_4 \in C_2 \cap C_3$ and $h_3 \in C_2$. One of the two conditions between $u_1(h_2) \geq u_1(h_3)$ and $u_1(h_2) \geq u_1(h_4)$ must be fulfilled. Finally, h_2 is a realisation of a Nash equilibrium if and only if $u_2(h_2) \geq u_2(h_1)$ and either $u_1(h_2) \geq u_1(h_3)$ or $u_1(h_2) \geq u_1(h_4)$.

In the next sections we apply Theorem 3.4 to introduce new algorithms to identify the realisations of the Nash equilibria of the game. Before getting to it, we state a property of the graph of the game, which will be helpful in the following arguments.

LEMMA 3.5 (TRIANGLE PROPERTY). *Given a game in its graph form $\langle H, I, u \rangle$ let us consider three outcomes $h, h', h'' \in H$. If $I(h, h'') \neq I(h'', h')$, then either $I(h, h') = I(h, h'')$ or $I(h, h') = I(h'', h')$.*

Example. Let us consider three vertices in the graph of Fig. 2b, for instance vertices h_1, h_2 and h_3 . The arcs (h_1, h_3) and (h_2, h_3) have label 1, while the arc (h_1, h_2) has label 2. The outcome h_3 is separated at the root by h_1 and h_2 and therefore shares with them the same label $i = 1$. Being the separation held at the same stage, h_3 must share the same label with h_1 and h_2 . With a similar argument it is possible to show that among three outcomes there is always one which is separated by the other two at the same stage. Formally, since it is impossible that three paths in a tree share three different intersections, it is possible to prove that no triangle in the graph of a game has three different labels.

4 ITERATIVE ALGORITHM

In this section we introduce a new method, namely Algorithm 4, to determine on the graph of the outcomes of the game, from now on also *game-graph*, whether a tagged outcome is a realisation of a Nash equilibrium. By iteration of Algorithm 4 on every outcome of the game, it is possible to enumerate the realisations of Nash equilibria.

The graph form of a game labels every pair of outcomes with the unique player who can be decisive in choosing among them. In order for an outcome $h \in H$ to be the realisation of a Nash equilibrium (cf. Theorem 3.4), any possible outcome $h' \in H \setminus \{h\}$ resulting from a deviation in terms of strategies must be either not incentivised, i.e., $u_i(h) \geq u_i(h')$, or not be a realisation of an unilateral deviation, i.e., there are at least two players having strategies that do not include h' as possible outcome. The first condition is easily verified by checking the values of the utility function for each tagged player i . Let us focus on the second condition: for any other outcome $h' \neq h$ with $I(h, h') = i$ we have to find sets $\{C_j\}_{j \in \mathcal{I} \setminus \{i\}}$ inducing maximal cliques on the respective graphs $\{\langle H|_{E_{\neq j}} \rangle\}_{j \in \mathcal{I} \setminus \{j\}}$ such that for at least one $j \in \mathcal{I} \setminus \{i\}$ we have that $h' \notin H(s_j)$. The aim of the Algorithm 4 is to verify the existence a set of maximal cliques such to prevent that any outcome not meeting the first condition $u_i(h) \geq u_i(h')$ at some player i belongs to the intersection of all cliques but the one corresponding to player i . Algorithm 4 allows, given a graph-game $\langle H, I, u \rangle$, precisely to determine if an outcome $h \in H$ is the realisation of a Nash equilibrium. In the following paragraphs we develop the steps that lead to the design of the algorithm:

- The graph $\langle H, E \rangle$ is partitioned into subgraphs $\{\langle H_i, E \rangle\}_{i \in \mathcal{I}}$, where $H = \cup_{i \in \mathcal{I}} \{H_i\} \cup \{h\}$ and $H_i = \{h' \in H, I(h, h') = i\}$ are the outcomes of the possible unilateral deviations of player i ;
- Any set $C_j \subset H$ inducing a maximal clique over $\langle H, E|_{\neq j} \rangle$ is shown to be a union of sets $C_j = \cup_{i \in \mathcal{I} \setminus \{j\}} C_j|_{H_i}$ inducing maximal cliques over the respective subgraphs $\{\langle H_i, E|_{\neq j} \rangle\}_{i \in \mathcal{I} \setminus \{j\}}$ (cf. Lemma 4.1); the problem can be thus analysed on every subgraph $\langle H_i, E|_{\neq j} \rangle$;
- The problem of existence of multiple sets $\{C_j|_{H_i} \subset H_i\}_{j \in \mathcal{I} \setminus \{i\}}$ inducing maximal cliques over the respective subgraphs $\{\langle H_i, E|_{\neq j} \rangle\}_{j \in \mathcal{I} \setminus \{i\}}$ is proved to be equivalent to the problem of existence of a set $C_i \subset H_i$ inducing a maximal clique over $\langle H_i, E|_{=i} \rangle$ (cf. Lemma 4.2);

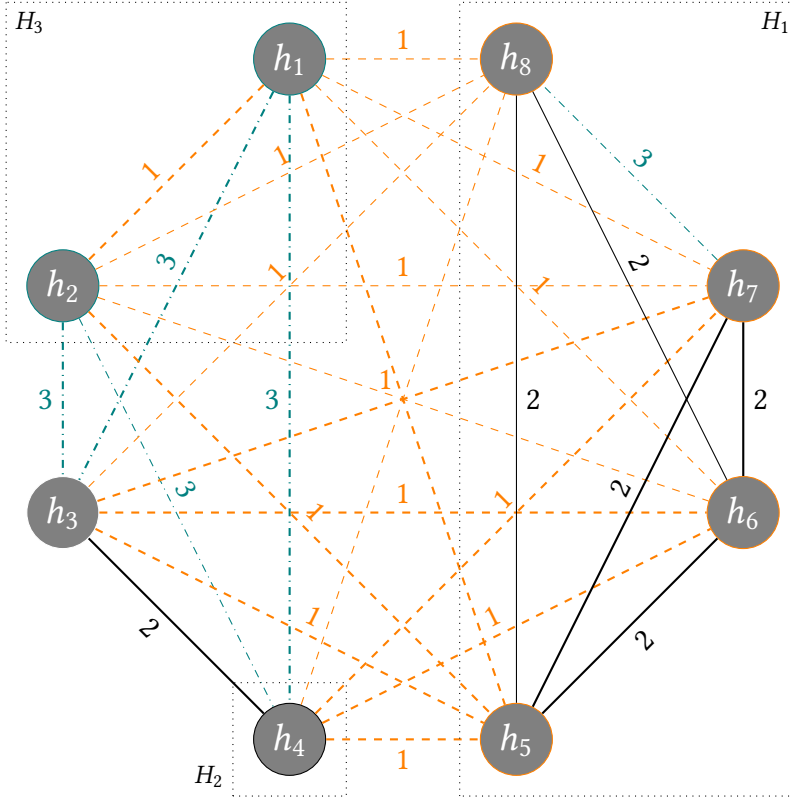


Fig. 3. **Graph form of game of Fig. 1.** Given $h = h_3$ as designated outcome, we have $H_1 = \{h_5, h_6, h_7, h_8\}$, $H_2 = \{h_4\}$ and $H_3 = \{h_1, h_2\}$. In the example the chosen strategy profile $s \in S$ is such that $H(s_1) = C_1 = \{h_1, h_3, h_4\}$, $H(s_2) = C_2 = \{h_1, h_2, h_3, h_5\}$ and $H(s_3) = C_3 = \{h_3, h_4, h_5, h_6, h_7\}$. Preferences of the players over the outcomes are respectively: $u_1 : h_6 >_1 h_7 >_1 h_8 >_1 h_3 >_1 h_4 >_1 h_2 >_1 h_1 >_1 h_5$, $u_2 : h_5 >_2 h_8 >_2 h_7 >_2 h_6 >_2 h_2 >_2 h_3 \sim_2 h_4 >_2 h_1$ and $u_3 : h_8 >_3 h_7 >_3 h_6 >_3 h_2 >_3 h_5 >_3 h_3 >_3 h_1 >_3 h_4$.

- Algorithm 4 thus checks for all $i \in I$ that on every subgraph $\langle H_i, E|_{=i} \rangle$ there is a set $C_i \subset H_i$ inducing a maximal clique such that none of the elements that do not meet the first condition $X_i = \{h' \in H_i, u_i(h) < u_i(h')\}$ belong to C_i , i.e., such that $C_i \cap X_i = \emptyset$ (cf. problem of the excluding maximal clique Algorithm 2);
- At every iteration Algorithm 4 solves a problem of the excluding clique (cf. Algorithm 3), rather than the equivalent problem of the excluding maximal clique (cf. Algorithm 2).

Induced subgraphs of deviations. From now on the object of the inquire is a designated outcome $h \in H$ and whether or not it is the realisation of some Nash equilibrium $s \in S$. For the sake of example we shall use the game-graph of Fig. 3 which corresponds to the game of Fig. 1. Let us group all the outcomes that can be potential unilateral deviations of the same player. In the following we call alternatively *unilateral deviation* the strategy $s'_i \neq s_i$ that differs from the one used at the Nash equilibrium $s \in S$ and the realisation h' of the new strategy profile $s' = (s'_i, s_{-i}) \mapsto h'$. Let us define the set of potential deviations of a given player $i \in I$ from h

$$H_i = \{h' \in H \setminus \{h\}, i = I(h, h')\}.$$

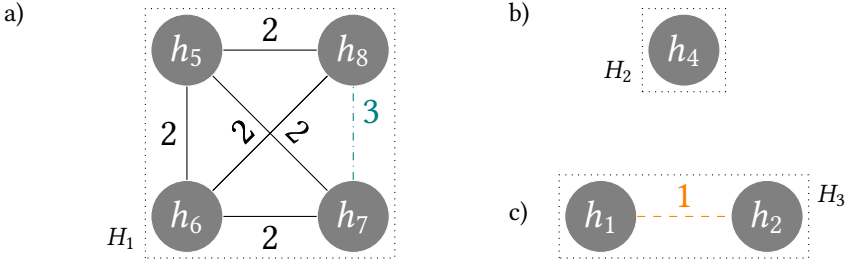


Fig. 4. **Induced subgraphs.** The candidate outcome is $h = h_3$. We show graph of Fig. 3 induced over a) $H_1 = \{h_5, h_6, h_7, h_8\}$ b) $H_2 = \{h_4\}$ and c) $H_3 = \{h_1, h_2\}$. We recall that the chosen $s \in S$ in the example is such that $H(s_1) = C_1 = \{h_1, h_3, h_4\}$, $H(s_2) = C_2 = \{h_1, h_2, h_3, h_5\}$ and $H(s_3) = C_3 = \{h_3, h_4, h_5, h_6, h_7\}$.

Example. In the game of Fig. 1 let us choose $h = h_3$ and verify whether or not it is a possible realisation of a Nash equilibrium. The possible deviations from h_3 for players 1, 2 and 3 are respectively $H_1 = \{h_5, h_6, h_7, h_8\}$, $H_2 = \{h_4\}$ and $H_3 = \{h_1, h_2\}$ (cf. Fig. 3). We split the set of possible deviations $H_i = V_i \cup X_i$ between those with no incentive of deviating $V_i = \{h' \in H_i, u_i(h) \geq u_i(h')\}$ and those whose utility would increase for the player $X_i = \{h' \in H_i, u_i(h) < u_i(h')\}$. Such division helps us to distinguish those elements, i.e., all the outcomes in $\{V_i\}_{i \in \mathcal{I}}$, that already meet the first condition of Theorem 3.4. In the aforementioned example we have $V_1 = \{h_5\}$, $X_1 = \{h_6, h_7, h_8\}$, $V_2 = \{h_4\}$, $X_2 = \emptyset$, $V_3 = \{h_1\}$ and $X_3 = \{h_2\}$.

Direct application of the main theorem. Let us apply Theorem 3.4 on the graph of Fig. 3 for $h = h_3$ by analysing every potential deviation $h' \in H_i$ for every $i \in \mathcal{I}$. Let us consider $C_1 = \{h_1, h_3, h_4\}$, $C_2 = \{h_1, h_2, h_3, h_5\}$ and $C_3 = \{h_3, h_4, h_5, h_6, h_7\}$ inducing maximal cliques respectively on $\langle H, E|_{\neq 1} \rangle$, $\langle H, E|_{\neq 2} \rangle$ and $\langle H, E|_{\neq 3} \rangle$. The elements in V_1 , V_2 and V_3 already meet the first condition of Theorem 3.4. Let us thus consider first $X_1 = \{h_6, h_7, h_8\}$ and then $X_3 = \{h_2\}$ (X_2 is empty). By construction, the elements in X_1 cannot belong to C_1 . Moreover, we find out that in this case none of the outcomes of X_1 neither belongs to C_2 . Therefore we can conclude that for $i = 1$ and every $h' \in X_1$ there is $j = 2$ such that $h' \notin C_2$. It is thus possible to define strategies such that the deviation to any element of X_1 is at least bilateral (both players 1 and 2 should change their strategies). Similarly, $h_2 \in X_3$ cannot belong to C_3 because it is player $I(h_2, h_3) = 3$ that can potentially deviate from h_3 to h_2 . Moreover, $h_2 \notin C_1$ and therefore player 1 guarantees that the deviation is bilateral, or formally for $i = 3$ and $h_2 \in C_3$ there is $j = 1$ such that $h_2 \notin C_1$. Summing up, we admit that outcomes h_1 , h_4 and h_5 cannot result from unilateral deviations, because they belong to respectively to V_3 , V_2 and V_1 and therefore there is no incentive for players to deviate to them. On the other hand, there exists a strategy profile such that all outcomes h_2 , h_6 , h_7 and h_8 cannot result from unilateral deviations, since they require at least a strategy deviation from at least two players simultaneously. Finally, Theorem 3.4 ensures that $h = h_3$ is the realisation of a Nash equilibrium of the game.

In order to simplify the condition of Theorem 3.4, we analyse the relationship between maximal cliques and the subgraphs induced by sets of possible deviations $\{H_i\}_{i \in \mathcal{I}}$.

Example. Fig. 4 shows the graph of Fig. 3 induced over respectively $H_1 = \{h_5, h_6, h_7, h_8\}$, $H_2 = \{h_4\}$ and $H_3 = \{h_1, h_2\}$. Let us suppose that we are given for all $j \in \mathcal{I}$ the set of outcomes C_j of a strategy $s_j \in S_j$ inducing a maximal clique over graph $\langle H, E|_{\neq j} \rangle$.

The first observation is that, in order for the designated $h \in H$ to be an outcome of the strategy profile $s \in S$, it must hold $h \in C_j$ for every $j \in \mathcal{I}$. By definition of H_j , we have that $C_j \cap H_j = \emptyset$, because all of its elements are incompatible with h with respect to player i .

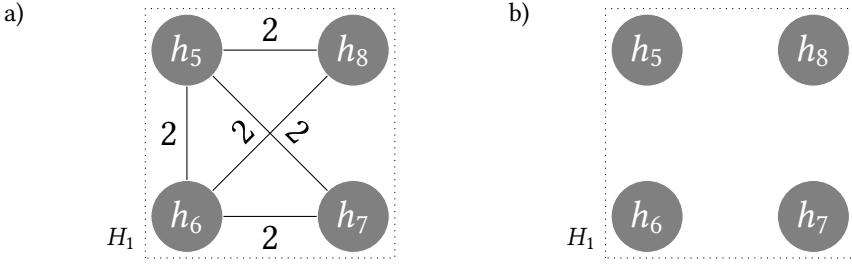


Fig. 5. **Removing arcs from a induced graph.** a) Induced graph $\langle H_1, E|_{\neq 3} \rangle$ b) Induced graph $\langle H_1, E|_{=1} \rangle$.

Example. In Fig. 4 we observe $C_1 \cap H_1 = \emptyset$, $C_2 \cap H_2 = \emptyset$ and $C_3 \cap H_3 = \emptyset$. Any set C_j that verifies the assumptions of Theorem 3.4 includes solely elements of outcome sets H_i for $i \neq j$.

The second observation is that the elements in $C_j \cap H_i$ for any $i \neq j$ form a maximal clique also on the induced graph $\langle H_i, E|_{\neq j} \rangle$, i.e., with H_i as set of vertices and $E|_{\neq j} = \{(h, h') \in H_i^2 \mid I(h, h') \neq j\}$.

Example. Let us consider $C_3 = \{h_3, h_4, h_5, h_6, h_7\}$ and let us analyse $C_3 \cap H_1$ and $C_3 \cap H_2$. The element h_4 is the only element of $C_3 \cap H_2$ and therefore forms a maximal clique within H_2 . The elements $\{h_5, h_6, h_7\} \subset C_3$ belong to $H_1 = \{h_5, h_6, h_7, h_8\}$. Let us observe the induced graph $\langle H_1, E|_{\neq 3} \rangle$ in Fig. 5a. The outcomes h_7 and h_8 are not connected since $I(h_7, h_8) = j = 3$. The elements $C_3 \cap H_1 = \{h_5, h_6, h_7\}$ form indeed a maximal clique within $\langle H_1, E|_{\neq 3} \rangle$.

With the following lemma we show that it is equivalent to look for a maximal clique over the graph $\langle H, E|_{\neq j} \rangle$ and looking for $N - 1$ maximal cliques on the $N - 1$ respective graphs $\{\langle H_i, E|_{\neq j} \rangle\}_{i \in I \setminus \{j\}}$.

LEMMA 4.1 (PARTITION). *Given a game in its graph form $\langle H, I, u \rangle$, a player $i \in I$ and the graph $\langle H, E|_{\neq j} \rangle$, every set C_j inducing a maximal clique over the graph $\langle H, E|_{\neq j} \rangle$ is the union $C_j = \cup_{i \in I \setminus \{j\}} C_j|_{H_i}$ of the disjoint sets $C_j|_{H_i} = C_j \cap H_i$ inducing maximal cliques over $\langle H_i, E|_{\neq j} \rangle$.*

PROOF. We recall that $H_i = \{h' \in H \mid I(h, h') = i\}$ is the set of possible deviations from the designated outcome $h \in H$. It is enough to prove that within the main graph $\langle H, E|_{\neq j} \rangle$ every two elements $h_{i'}$, $h_{i''}$ belonging to two different sets of possible deviations, i.e., $h_{i'} \in H_{i'}$ and $h_{i''} \in H_{i''}$ with $i', i'' \in I \setminus \{j\}$ and $i' \neq i''$, are always connected. Formally, we need to show that it always holds $(h_{i'}, h_{i''}) \in E|_{\neq j}$, i.e., $I(h_{i'}, h_{i''}) \neq j$. Given the designated $h \in H$, we observe that $I(h, h_{i'}) = i'$ and $I(h, h_{i''}) = i''$. For the triangle property of Lemma 3.5 either $I(h_{i'}, h_{i''}) = i' \neq j$ or $I(h_{i'}, h_{i''}) = i'' \neq j$, which concludes the proof. \square

Example. Let us consider again the game-graph of Fig. 3 with candidate outcome $h = h_3$. Let us characterise a generic maximal clique C_2 , i.e., the set of outcomes of a strategy $s_2 \in S_2$ of player $j = 2$ that admits $h_3 \in H(s_2) = C_2$ as possible outcome. By hypothesis we have to fix $h_3 \in C_2$ and $h_4 \notin C_2$, since $H_2 = \{h_4\}$. Let us consider thus $H_1 = \{h_5, h_6, h_7, h_8\}$ and $H_3 = \{h_1, h_2\}$. Given any edge $(h', h'') \in H_1 \times H_3$, the label $I(h', h'') \neq 2$ again from Lemma 3.5. In this specific case $I(h', h'') = 1$ for every pair of elements. Any candidate strategy $s_2 \in S_2$ for a Nash equilibrium $s \in S$ having $s \mapsto h_3$ as realisation has therefore a set of outcomes $H(s_2) = C_2 = \{h_3\} \cup (C_2|_{H_1}) \cup (C_2|_{H_3})$, where $C_2|_{H_1}$ and $C_2|_{H_3}$ are sets of elements that induce a maximal clique respectively on $\langle H_1, E|_{\neq 2} \rangle$ and $\langle H_3, E|_{\neq 2} \rangle$.

Theorem 3.4 requires to identify for every $j \in I$ a maximal clique over the graph $\langle H, E|_{\neq j} \rangle$. Thanks to the latest result, it is possible to check the existence of such maximal clique on every induced graph $\langle H_i, E|_{\neq j} \rangle$. We thus rewrite the necessary and sufficient condition on the induced subgraphs: for all $i \in I$ there must be a set $C_j|_{H_i}$ inducing a maximal clique for every player

ALGORITHM 2: (MC) Existence of a maximal clique excluding a set of vertices

Input: $\langle H, E, X \rangle$ defining a graph $\langle H, E \rangle$ and a subset of vertices $X \subset H$.

Output: Is there a vertex set $C \subset H \setminus X$ that induces a maximal clique on $\langle H, E \rangle$?

$j \in \mathcal{I} \setminus \{i\}$ such that none of the possible deviations $h' \in X_i \subset H_i$ belongs to the intersection of the maximal cliques $\bigcap_{j \in \mathcal{I} \setminus \{i\}} C_j|_{H_i}$.

Example. In Fig. 5a we have that $X_1 = \{h_6, h_7, h_8\}$ and the induced sets over $H_1 = \{h_5, h_6, h_7, h_8\}$ that induce maximal cliques are respectively $C_2|_{H_1} = \{h_5\}$ and $C_3|_{H_1} = \{h_5, h_6, h_7\}$. The property $(\bigcap_{j \in \mathcal{I} \setminus \{i\}} C_j) \cap X_i = \emptyset$ is fulfilled, since $(C_2 \cap C_3)|_{H_1} = \{h_5\}$ and $X_1 = \{h_6, h_7, h_8\}$ have no elements in common.

Let us discuss the properties of the intersection $\bigcap_{j \in \mathcal{I} \setminus \{i\}} C_j|_{H_i}$: these are the possible unilateral deviations of player i , given the strategies of the other players $\bigcap_{j \in \mathcal{I} \setminus \{i\}} C_j|_{H_i} = H(s_{-i})$ for some $s_{-i} \in S_{-i}$. Let us observe that given the possible outcomes $\bigcap_{j \in \mathcal{I} \setminus \{i\}} C_j|_{H_i}$ it is the player $i \in \mathcal{I}$ who chooses which one is to be the deviation. In other words, it is intuitive that for every pair of elements $h, h' \in \bigcap_{j \in \mathcal{I} \setminus \{i\}} C_j|_{H_i}$ it must hold $I(h, h') = i$. Let us show that this property is maximal and thus that identifying an intersection of maximal cliques $\bigcap_{j \in \mathcal{I} \setminus \{i\}} C_j|_{H_i}$ is equivalent to identifying a set C_i inducing a maximal clique over H_i .

LEMMA 4.2. *Given a game in graph form $\langle H, I, u \rangle$, a player $i \in \mathcal{I}$ and the induced subgraph $\langle H_i, E \rangle$ on the set of possible deviations H_i and a set $C_i \subset H_i$, the two conditions are equivalent:*

- C_i induces a maximal clique over graph $\langle H_i, E|_{=i} \rangle$, where $E|_{=i} = \{(h, h') \in H_i^2 | I(h, h') = i\}$;
- There are sets $\{C_j|_{H_i}\}_{j \in \mathcal{I} \setminus \{i\}}$ inducing maximal cliques over the graphs $\{\langle H_i, E|_{\neq j} \rangle\}_{j \in \mathcal{I} \setminus \{i\}}$ such that $C_i = \bigcap_{j \in \mathcal{I} \setminus \{i\}} C_j|_{H_i}$.

PROOF. For sake of clarity, in the proof we drop the subscript $|_{H_i}$ from $C_j|_{H_i}$.

We first prove the direct implication. Let C_i induce a maximal clique over $\langle H_i, E|_{=i} \rangle$ and for all $j \in \mathcal{I} \setminus \{i\}$ a set C_j with $C_i \subset C_j$ induce maximal clique over the graphs $\langle H_i, E|_{\neq j} \rangle$. The sets $\{C_j\}$ are well defined because for all $h', h'' \in C_i$ it holds $I(h', h'') = i \neq j$. We observe that for every $h', h'' \in \bigcap_j C_j$ it holds $I(h', h'') = i$ and thus $C_i \subset \bigcap_j C_j$. Since C_i is a maximal clique, $C_i = \bigcap_j C_j$.

Now let us prove the opposite, i.e., let us show that $C_i = \bigcap_j C_j$ induces a maximal clique over $\langle H_i, E|_{=i} \rangle$. For all $h', h'' \in C_i$ it holds $I(h', h'') = i$, i.e., C_i induces a clique on $\langle H_i, E|_{=i} \rangle$. Let us show that it is maximal. By contradiction, there is $C'_i \supset C_i$ forming a maximal clique over $\{H_i, E|_{=i}\}$. With a similar argument used for Lemma 3.3 we prove that this is absurd. \square

Let us observe the induced graph $\langle H_1, E|_{=1} \rangle$ of Fig. 5b. There are no edges with label 1 and therefore all the maximal cliques are the single vertices. This means that players 2 and 3 can identify strategies such that player 1 is forced to pick only the vertex chosen by them.

Problem of the excluding maximal clique. The necessary and sufficient condition of Theorem 3.4 is equivalent to verify whether on every graph $\langle H_i, E|_{=i} \rangle$ there is a set C_i that induces a maximal clique such that $C_i \cap X_i = \emptyset$. The problem is known as the the existence of a maximal clique excluding a set of vertices (cf. Algorithm 2).¹ The condition is thus to verify with the problem of Algorithm 2 with input $\langle H, E, X \rangle = \langle H_i, E|_{=i}, X_i \rangle$ for every $i \in \mathcal{I}$ the existence of a set $C \subset H_i \setminus X_i$ inducing a maximal clique over H_i .

Equivalent problem of the excluding clique. In [Zappalà et al., 2023] it is proved that such problem is equivalent to determine whether there is a (non-maximal) clique on $V = H \setminus X$ such that for every element in X there is at least one in V not connected to it (cf. Algorithm 3). Such problem is

¹For sake of clarity, the definitions of the problem of the excluding maximal clique and the problem of the excluding clique [Zappalà et al., 2023] are presented in the Appendix A.

ALGORITHM 3: (EC) Existence of a clique excluding a set of vertices

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

Output: Is there a vertex set $C \subset V$ that induces a clique on $\langle V \cup X, E \rangle$ such that for all $h'' \in X$ there is $h' \in C$ so that $(h', h'') \notin E$?

ALGORITHM 4: (NE) Determining whether an outcome is a realisation of a Nash equilibrium

Input: A game in its graph form $= \langle H, I, u \rangle$ and an outcome $h \in H$.

Output: Is h a realisation of a Nash equilibrium?

 $boolean = True;$ // A boolean value determining whether h is a realisation of a NE

for $i \in \mathcal{I}$ **do**
 $V_i \leftarrow \{h' \in H \setminus \{h\}, I(h, h') = i, u_i(h) \geq u_i(h')\};$
 $X_i \leftarrow \{h' \in H \setminus \{h\}, I(h, h') = i, u_i(h) < u_i(h')\};$
 $E|_{V_i \cup X_i} = \{(h, h') \in V_i \cup X_i, I(h, h') = i\};$
if *Output of Algorithm 3 with input $\langle V_i, X_i, E|_{V_i \cup X_i} \rangle$ is negative* **then**
 $boolean = False;$
end
end

NP-complete on a generic graph, but it comes with a simple integer linear formulation [Zappalà et al., 2023].

We can thus rely on Theorem 3.4, the latest results and the newly introduced problem to present Algorithm 4 which determines whether an outcome is a realisation of a Nash Equilibrium.

THEOREM 4.3. *Given a game in its graph form $\langle H, I, u \rangle$ and an outcome $h \in H$ as input, Algorithm 4 determines whether h is a realisation of a Nash equilibrium.*

PROOF. Let us define the set of possible unilateral deviations $H_i = \{h' \in H, I(h, h') = i\}$, with $H_i = V_i \cup X_i$ as in Algorithm 4. Theorem 3.4 states that the necessary and sufficient condition for h to be a realisation of a Nash equilibrium is the existence of sets $\{C_i\}_{i \in \mathcal{I}}$ inducing maximal cliques over the respective graphs $\{\langle H, E|_{\neq i} \rangle\}_{i \in \mathcal{I}}$ such that $h \in \bigcap_{i \in \mathcal{I}} C_i$ and $\bigcap_{j \in \mathcal{I} \setminus \{i\}} C_j \subset V_i$. The previous results let us simplify such condition up to the excluding clique problem: for sake of clarity, hereafter let us summarise the argument. For Lemma 4.1 the condition is equivalent to verifying for all $i \in \mathcal{I}$ the existence of sets $\{C_j|_{H_i} \subset H_i\}_{j \in \mathcal{I} \setminus \{i\}}$ inducing maximal cliques over graphs $\{\langle H_i, E|_{\neq j} \rangle\}_{i \in \mathcal{I}}$ such that $\bigcap_{j \in \mathcal{I} \setminus \{i\}} C_j|_{H_i} \subset V_i$. Finally, for Lemma 4.2 this condition is equivalent to verifying for all $i \in \mathcal{I}$ the existence of a set $C_i \subset V_i$ that induces a maximal clique over $\langle H_i, E|_{=i} \rangle$. Since the problem of the excluding maximal clique is equivalent to the problem of the excluding clique [Zappalà et al., 2023], we have the proof. \square

5 RECURSIVE ALGORITHM

Algorithm 4 allows us to determine if an outcome is a possible realisation of a Nash equilibrium. If we iterate the algorithm over all the outcomes of the game, it is possible to enumerate all the realisations of Nash equilibria. However, we observe that many computations are repeated. In this section we discuss Algorithm 5, that allows to perform the enumeration while avoiding such repetitions. The following arguments are developed in the rest of the section:

- If we fix the graph $\langle H_i, E|_{=i} \rangle$, the problem of identifying a clique on H_i excluding $X_i = \{h' \in H_i, u_i(h) < u_i(h')\}$ depends solely on the parameter $u_i(h)$. Not all designated outcomes h allow to find an excluding clique: there is a value $\bar{u} = u_i(\theta_i)$ for some $\theta_i \in H_i$ over which the

clique exists and under which the clique does not; we call this value *threshold* outcome (cf. Definition 5.1);

- Solving the problem of the excluding clique for a given $h \in H$ on a graph $\langle H_i, E|_{=i} \rangle$ is thus equivalent to comparing the value of $u_i(h)$ to $u_i(\theta_i)$, i.e., the utility of the threshold outcome $\theta_i \in H_i$; Algorithm 5 avoid repetitions by first computing such thresholds;
- The set H_i is the union of different subgames and thus its threshold can be obtained by either maximising or minimising the utility of the thresholds of such subgames (cf. Lemma 5.2);
- It is thus possible to compute all thresholds recursively on the game-tree by decomposing every subgames in smaller ones;
- Algorithm 5 recursively computes the thresholds of every subgame and checks simultaneously whether every outcome $h \in H$ meets them (cf. Theorem 5.3).

Example. Let us consider the game-graph of Fig. 3. Let us apply Algorithm 4 to the designated outcomes $h_3, h_4 \in H$. In the case of $h_3 \in H$ we recall that the possible unilateral deviations for the players are respectively $H_1 = \{h_5, h_6, h_7, h_8\}$, $H_2 = \{h_4\}$ and $H_3 = \{h_1, h_2\}$. For the case of outcome $h_4 \in H$ by inspection the possible deviations are $H_1 = \{h_5, h_6, h_7, h_8\}$, $H_2 = \{h_3\}$ and $H_3 = \{h_1, h_2\}$. Even if H_1 and H_3 are unchanged, the input of Algorithm 2, namely $\langle H_i, E|_{\neq i}, X_i \rangle$, with $X_i = \{h' \in H_i | u_i(h) < u_i(h')\}$, is changed. However, the only parameter that changes is $u_i(h)$.

We can avoid to repeat part of such computations by identifying the lowest value of utility $u_i(h)$ that an outcome $h \in H$ must achieve in order to admit a set C_i inducing a maximal clique over $\langle H_i, E|_{=i} \rangle$ such that $C_i \cap X_i = \emptyset$, with $X_i = \{h' \in H_i, u_i(h) < u_i(h')\}$. Let us observe that if $X_i = \emptyset$ for some $u_i(h)$, then the maximal clique always exists. As the parameter $u_i(h)$ decreases, the set X_i gets larger and it is thus less likely to find a maximal clique. When $u_i(h)$ is lower than the value of $u_i(h')$ for any $h' \in H_i$, we have that $X_i = H_i$ and thus the maximal clique does not exist. The intuition is the following: the lowest value of utility that an outcome can attain corresponds to an element of H_i which is the one with the smallest utility among those which that cannot be included in X_i for a maximal clique to be identified in $H_i \setminus X_i$. For this reason, we call this element a *threshold* outcome.

Definition 5.1 (Threshold). Given a game in graph form $\langle H, I, u \rangle$, a player $i \in I$, a designated outcome h and the set $H_i \subset H$ of her possible deviations, a threshold outcome $\theta_i \in H_i$ is such that Algorithm 2 with input $\langle H_i, E|_{\neq i}, X_i \rangle$ with $X_i = \{h' \in H_i | \bar{u} < u_i(h')\}$ admits solutions for every $\bar{u} \geq u_i(\theta_i)$ and does not admit solution for every $\bar{u} < u_i(\theta_i)$.

It is obvious that player i attains same utility at threshold outcomes.

Example. In the game-graph of Fig. 3 the corresponding subgraphs $\langle H_1, E|_{=1} \rangle$, and $\langle H_3, E|_{=3} \rangle$ have trivially no edges, and thus every outcome alone generates a maximal clique. In this case the minimum value admitted for h_3 and h_4 are the minimum values of the elements of the induced graphs in H_1 for player 1 and in H_3 for player 3. Such values, which are the thresholds for H_1 and H_3 , are respectively h_5 and h_1 : $u_1^{min} = \min_{h' \in H_1} u_1(h') = u_1(h_5)$ and $u_3^{min} = \min_{h' \in H_3} u_3(h') = u_3(h_1)$. The candidate outcomes h_3 and h_4 must be compared first with $\theta_1 = h_5$ with respect to player 1: we observe both $h_3 \geq_1 h_5$ and $h_4 \geq_1 h_5$. Then they both must be compared to $\theta_3 = h_1$ with respect to player 3: we have that $h_3 \geq_3 h_1$, but $h_4 <_3 h_1$. Therefore h_4 cannot be the realisation of a Nash equilibrium, because no maximal clique can be identified on H_3 , indeed $X_3 = H_3$.

The first intuition at the base of Algorithm 5 is that that the set of possible deviations H_i from a target outcome h is the union of the outcomes of all the subgames that must be avoided by player i in order to attain h . Furthermore, such subgames are separated one from the other by the label i on the game-graph, because it is player i who is decisive in not attaining them.

Example. Let us go back to the game-tree of Fig. 1: in order to get to h_3 or h_4 player 1 does not go right at the root. Note that the corresponding subgame is identified by the deviating

outcomes $H_1 = \{h_5, h_6, h_7, h_8\}$, so that we can associate to it the corresponding threshold outcome h_5 corresponding to H_1 . Then, again, at the following node, player 3 need not to go left. By direct computation, the threshold for the corresponding subgame, identified this time by the deviating outcomes $H_3 = \{h_1, h_2\}$, is h_1 . Note that, in this case, the sets H_1 and H_3 correspond to two unique subgames.

The second intuition of Algorithm 5 is that in order to prevent a deviation of player $i \in \mathcal{I}$ from h to another outcome belonging to a subgame, the other players pick strategies so as to let her available the outcomes with the lowest utility. Among these outcomes player i chooses the one with the highest utility h' . If this outcome is preferred to h' by player i , there is no way for the other players to keep her from deviating to the subgame. The outcome h' with the lowest value of utility that can be chosen by player $i \in \mathcal{I}$ is later proved to be the threshold of the subgame rooted at the node where player i acts. More specifically, we show below that the threshold outcome is obtained via a *minmax* argument: it is obtained by supposing that in every subgame all players but i minimise the value of the utility function u_i , while player i maximises it. Hence, this argument is the basis for the design of a recursive computation of thresholds and equilibria on the game-tree. The key observation is that the game-graph of a subgame can be partitioned in the game-graphs of the subgames at the lowest stage; such subgames are separated by the same label $j \in \mathcal{I}$ of the player who chooses among them at the stage before.

Example. In Fig. 1 in the subgame generated by outcomes $H_1 = \{h_5, h_6, h_7, h_8\}$ players 2 and 3 identify a combination of strategies leading to h_5 , which minimises the utility of the player $i = 1$. The subgames $\{h_5\}$, $\{h_6\}$ and $\{h_7, h_8\}$ are all connected by label $j = 2$ (cf. Fig. 3).

In order to prove formally these two arguments, we need a method that allows to analyse a game-graph when it is partitioned in sets separated by the same label. Such decomposition method is justified by the following lemma.

LEMMA 5.2 (DECOMPOSITION). *Let us consider a game in this graph form $\langle H, I, u \rangle$, a player $i \in \mathcal{I}$, a set of possible deviations $H_i = \{h' \in H \setminus \{h\}, I(h, h') = i\}$ and one of its partition $\{H_i^k\}_{k \in K}$ such that for every $k', k'' \in K$ and for every pair of outcomes $h_{k'} \in H_i^{k'}$ and $h_{k''} \in H_i^{k''}$ with $k' \neq k''$ it holds $I(h_{k'}, h_{k''}) = j \in \mathcal{I}$. Let us consider the respective thresholds $\theta_i \in H_i$ and $\{\theta_i^k \in H_i^k\}_{k \in K}$, then*

- if $j = i$, $\theta_i \in \arg \max_{k \in K} u_i(\theta_i^k)$;
- if $j \neq i$, $\theta_i \in \arg \min_{k \in K} u_i(\theta_i^k)$.

PROOF. Let us consider the case $j = i$. The proof is based on the fact that every clique on $H_i \setminus X_i$ is a union of cliques over $H_i^k \setminus X_i^k$. It can be thus be proved that if we fix $h \in H \setminus \{H_i\}$ such that $u_i(h) = \max_{k \in K} u_i(\theta_i^k)$ the Algorithm 3 with input $\langle H_i, E|_{\neq i}, X_i \rangle$, where $X_i = \{h' \in H_i, u_i(h) < u_i(h')\}$ always admits solution. On the other hand, if we fix $h \in H \setminus \{H_i\}$ such that there is $\theta' \in H_i$ with $u_i(h) \leq u_i(\theta') < \max_{k \in K} u_i(\theta_i^k)$ the Algorithm 3 does not admit solution, because there is at least one element in X_i^k for some $k \in K$ that is not excluded by the clique induced on H_i .

Let us consider the case $j \neq i$. The argument of the proof is analogous. We observe that the sets $\{H_i^k\}_{k \in K}$ are disconnected within $\langle H_i, E|_{=i} \rangle$. Therefore for any set $C \subset H_i \setminus X_i$ inducing a clique on H_i there is $k \in K$ and $C^k \subset H_i^k \setminus X_i^k$ such that $C = C^k$. The thresholds θ_i is computed on the same clique giving solution to the lowest of the $\{\theta_i^k\}_{k \in K}$. \square

The set of possible deviations H_i is the union of outcomes of the subgames separated from the tagged h by player i . Given two elements h', h'' belonging to two of these different subgames, we have that $I(h', h'') = i$. Therefore we apply Lemma 5.2 to compute the thresholds on every subgame: the maximum value of utility of these is the threshold for H_i .

Within a subgame Γ different players act. Let us note that at the root $h^0 \in H'$ the subgame can be further split in as many subgames Γ^k as the actions available in $\mathcal{A}(h^0)$ for $j = P(h^0)$. Every two outcomes $h^{k'}, h^{k''}$ belonging to two different $\Gamma^{k'} \neq \Gamma^{k''}$ have as label $I(h^{k'}, h^{k''}) = j$. We can thus apply Lemma 5.2 and first compute the threshold on every subgame before maximising (if $j = i$) or minimising (if $j \neq i$) the respective results for the function u_i . Recursively, we can thus split every subgame in smaller ones and compute the thresholds on them before obtaining the general threshold for Γ .

Recursive computation of subgame thresholds. Algorithm 5 adopts the same recursive structure of the backward induction (cf. Algorithm 1). In practice it computes the thresholds $\theta = (\theta_i)_{i \in \mathcal{I}}$ at every node starting from the leaves. If the game is degenerate, i.e., it has only one element $H = 1$, such element is the Nash equilibrium and it is also the unique threshold no matter whose player's utility is to be compared. In a generic game Algorithm 5 computes at each stage the thresholds $\{\theta_i \in H\}_{i \in \mathcal{I}}$ by picking for the player $i = P(h')$ acting at a node $h' \in H'$ the thresholds at the following stage that maximises her utility, while for the other players $j \in \mathcal{I} \setminus \{i\}$ the one that minimise their utility.

Example. Let us identify such thresholds for the game of Fig. 1. The computations are shown in Fig. 6. Let us start from the bottom nodes on the third stage. Player 1 must choose between h_1 and h_2 . Following Algorithm 5, we identify the outcome with largest utility for player 1 ($\theta_1 = h_2$, because $h_2 \succ_1 h_1$) and the lowest utility for the other players ($\theta_2 = \theta_3 = h_1$ in both cases, since $h_1 \prec_2 h_1$ and $h_1 \prec_3 h_2$). Player 2 chooses between h_3 and h_4 , which are equivalent for her $h_3 \sim_2 h_4$, therefore $\theta_2 = h_3 = h_4$ (since both can be chosen, for clarity in Fig. 6 we write only $\theta_2 = h_4$). For the other two players 1 and 3 it is necessary to minimise the utility: indeed, we have $\theta_1 = \theta_3 = h_4$, because $h_4 \prec_1 h_3$ and $h_4 \prec_3 h_3$. Player 3 has to choose between the h_7 and h_8 . Maximising for the utility of player 3 and minimising for the other two players ($h_8 \prec_1 h_7$, $h_7 \prec_2 h_8$ and $h_8 \succ_3 h_7$), we obtain $(\theta_1, \theta_2, \theta_3) = (h_8, h_7, h_8)$. Let us now compute the thresholds for the nodes at the second stage. Player 3 must choose between left and right, whose thresholds are respectively $\theta = (h_2, h_1, h_1)$ and $\theta = (h_4, h_4, h_4)$. We maximise for player 3 and thus have $\theta_3 = h_1$, because $h_1 \succ_3 h_4$, while we minimise for player 1 and 2, getting $\theta_1 = \arg \min_{u_1(\cdot)}(h_2, h_4) = h_2$ and $\theta_2 = \arg \min_{u_2(\cdot)}(h_1, h_4) = h_1$. At the second stage player 2 has three actions available (left, centre, right). Left and centre actions lead to outcomes of the game h_5 and h_6 and thus we fix for them the thresholds $\theta = (h_5, h_5, h_5)$ and $\theta = (h_6, h_6, h_6)$. The thresholds to be compared are thus $\theta = (h_5, h_5, h_5)$, $\theta = (h_6, h_6, h_6)$ and $\theta = (h_8, h_7, h_8)$. Let us maximise for the utility of player 2, getting $\theta_2 = \arg \max_{u_2(\cdot)}(h_5, h_6, h_7) = h_5$, and minimise for the utility of the other players 1 and 3, leading thus to $\theta_1 = \arg \max_{u_1(\cdot)}(h_5, h_6, h_8) = h_5$ and $\theta_3 = \arg \max_{u_3(\cdot)}(h_5, h_6, h_8) = h_5$. Finally at the root player 1 has to choose between left and right actions, whose thresholds are (h_2, h_1, h_1) and (h_5, h_5, h_5) . For player 1 we have $h_2 \succ_1 h_5$ and thus $\theta_1 = h_2$ (utility is maximised), while for players 2 and 3 we have $\theta_2 = \theta_3 = h_1$, since $h_5 \succ_2 h_1$ and $h_5 \succ_3 h_1$.

Comparison between outcomes and respective thresholds. Once the thresholds are computed at each stage, a backward propagation argument let us determine the outcomes who are realisation of a Nash equilibrium. In order for an outcome to be a realisation of a Nash equilibrium, it should avoid unilateral deviations at every stage from the root to the leaf. The threshold at every stage is the lowest value a unilateral deviation can achieve in a subgame which is not explored. If an outcome does not meet a threshold at a certain stage, it means that the player acting at the stage has an incentive to deviate unilaterally. Now that the thresholds are computed, we can thus verify which outcomes meet all the thresholds at every stage. In fact, Algorithm 5 performs this control while computing the thresholds (cf. Fig. 6): we will discuss this approach later, for the moment we perform the analysis outcome by outcome.

ALGORITHM 5: (EA) Enumeration Algorithm

Input: A game $\Gamma = \langle \mathcal{I}, \mathcal{A}, H', H, P, u \rangle$ and its root $h^0 = \emptyset \in H'$.

Output: $\langle NE, \langle \theta_i \rangle_{i \in \mathcal{I}} \rangle$, i.e. the set of outcomes which are realisations of all Nash equilibria $NE \subset H$ and the thresholds $\langle \theta_i \rangle_{i \in \mathcal{I}}$ for every player.

```

 $\langle NE, \langle \theta_i \rangle_{i \in \mathcal{I}} \rangle \leftarrow \emptyset$ ; // Initialising the output
if  $H = 1$  then
   $NE = H$ ; // degenerate game-tree (leaf)
   $\theta_i = h$  for all  $i \in \mathcal{I}$ , where  $H = \{h\}$ ; // is also a threshold outcome
else
   $\langle NE^k, \langle \theta_i^k \rangle_{i \in \mathcal{I}} \rangle_{a_k \in \mathcal{A}(h^0)} \leftarrow \emptyset$ ; // Output of the subgames of the root node
  for  $a_k \in \mathcal{A}(h^0)$  do
     $\Gamma^k = \Gamma(h^0 + (a_k))$ ; // The subgame that follows actions  $a_k$ 
     $\langle NE^k, \theta^k \rangle = EA(\Gamma^k)$ ; // The candidate outcomes and the thresholds
  end
  for  $i \in \mathcal{I}$  do
    if  $i = P(h^0)$  then
       $\theta_i = \arg \max_k u_i(\theta_i^k)$ ; // Maximising for the player acting at the node
       $NE = \{h \in \cup_k NE^k \mid h \geq_i \theta_i\}$ ;
    else
       $\theta_i = \arg \min_k u_i(\theta_i^k)$ ; // Minimising for the other players
    end
  end
end

```

Example. Let us start from h_1 : starting from the bottom to the root the thresholds are $\theta_1 = h_2$ (player 1 on the third stage), $\theta_3 = h_1$ (player 3 on the second stage) and $\theta_1 = h_2$ (player 1 at the root). The thresholds are highlighted in Fig. 6. The outcome h_1 does not meet two thresholds, indeed $h_1 <_1 \theta_1 = h_2$, $h_1 \sim_3 \theta_3 = h_1$ and $h_1 <_1 \theta_1 = h_2$. Therefore it is not the realisation of a Nash equilibrium. On the other hand, outcome h_2 has the same thresholds and it does meet them all: $h_2 \sim_1 \theta_1 = h_2$, $h_2 >_3 \theta_3 = h_1$ and $h_2 \sim_1 \theta_1 = h_2$. Therefore it is the realisation of a Nash equilibrium; indeed, we verified before (cf. Fig. 1b) that it is the outcome of the subgame perfect equilibrium. The outcomes h_3 and h_4 have the following thresholds $\theta_2 = h_4$, $\theta_3 = h_1$ and $\theta_1 = h_2$. Outcome h_3 meets all the thresholds: $h_3 \sim_2 \theta_2 = h_4$, $h_3 >_3 \theta_3 = h_1$ and $h_3 >_1 \theta_1 = h_2$. It is thus the realisation of a Nash equilibrium. Outcome h_4 fails to meet threshold $h_4 <_3 \theta_3 = h_1$, and therefore it is not a realisation of a Nash equilibrium. Outcome h_5 and h_6 have two thresholds $\theta_2 = h_5$ and $\theta_1 = h_2$. Outcome h_6 fails to meet the first one, $h_6 <_2 \theta_2 = h_5$, while outcome h_5 fails to meet the second one, $h_5 <_1 \theta_1 = h_2$. Finally outcome h_7 and h_8 have three thresholds: $\theta_3 = h_8$, $\theta_2 = h_5$ and $\theta_1 = h_2$. Both fail to meet $h_7 <_2 h_8 <_2 \theta_2 = h_5$. Therefore none of them is a realisation of Nash equilibria.

In Fig. 6 we observe that Algorithm 5 verifies if an outcome meets a thresholds right after computing them. Those potential outcomes of Nash equilibria NE^k that met all the thresholds in subgame Γ^k are tested on the upper stage of the game, recursively up to the root.

THEOREM 5.3. *Given a game $\Gamma = \langle \mathcal{I}, \mathcal{A}, H', H, P, u \rangle$ and its root $h^0 = \emptyset \in H'$ as input, Algorithm 5 provides all the possible realisations of a Nash equilibrium of the game.*

PROOF. Let us consider a tagged outcome $h \in H$ and the sets $H_i = \{h' \in H \mid I(h, h') = i\}$, $X_i = \{h' \in H_i \mid u_i(h) < u_i(h')\}$, $V_i = H_i \setminus X_i$. Theorem 4.3 shows that h is the realisation of a Nash equilibrium if and only if there exists clique $C_i \subset V_i$ on every graph $\langle H_i, E_{|=i} \rangle$. The definitions of

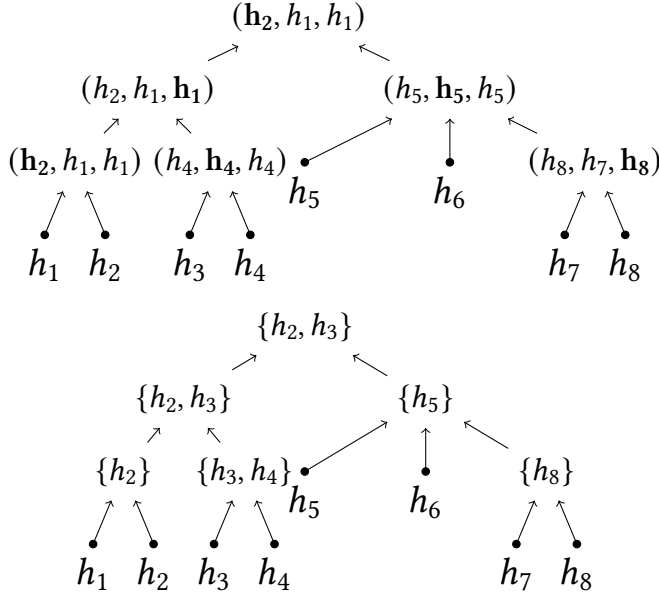


Fig. 6. **Application of Algorithm 5.** Game in extensive form. Preferences of the players over the outcomes are respectively: $u_1 : h_6 >_1 h_7 >_1 h_8 >_1 h_3 >_1 h_4 >_1 h_2 >_1 h_1 >_1 h_5$, $u_2 : h_5 >_2 h_8 >_2 h_7 >_2 h_6 >_2 h_2 >_2 h_3 \sim_2 h_4 >_2 h_1$ and $u_3 : h_8 >_3 h_7 >_3 h_6 >_3 h_2 >_3 h_5 >_3 h_3 >_3 h_1 >_3 h_4$. Above: Thresholds $\theta^k = (\theta_1^k, \theta_2^k, \theta_3^k)$ at every node. Below: Nash Equilibria NE^k of every subgame Γ^k .

thresholds (cf. Definition 5.1) allows to rewrite the problem as $u_i(h) \geq u_i(\theta_i)$ for all $i \in I$. Given any $h^k < h$ the k -prefix of h , θ_i^k the threshold of every subgame Γ^k not explored by player $i = P(h^k)$, Algorithm 5 guarantees that $u_i(h) \geq u_i(\theta_i^k)$ for every $h^k < h$. Since H_i is the union of all subgames not explored by player i , Lemma 5.2 implies that the two conditions are equivalent. \square

Complexity analysis. Differently from Algorithm 5, Algorithm 4 does not require to build the full game-tree to determine if an outcome is a realisation of a Nash equilibrium. It can be extremely efficient when the full enumeration is not required or when the sets of deviations $\{H_i\}_{i \in I}$ has low values of utility. Moreover, it is recommended when the game-tree is too large to be stored.

We have remarked before that when Algorithm 4 is used to iteratively compute every outcome of the Nash equilibria, it repeats the analysis of possibly the same sets of outcomes, and relies on the resolution of a problem which is NP-complete for a generic graph. On the the other hand, Algorithm 5 is more efficient, as it avoids every repetition and exploits the properties of the graph (cf. Lemma 3.5). Indeed, let us prove that Algorithm 5 is polynomial with respect to the size of the game, i.e., to the number of outcomes $|H|$ of the game. The recursive structure of Algorithm 5 makes it a generalisation of the backward induction (cf. Algorithm 1). We characterise the complexity of Algorithm 5 using similar arguments already used for the backward induction [Szymanik, 2013]. The result exploits the property that the intermediate histories $|H' \setminus H|$ are fewer than the outcomes $|H|$ and that the number of operations at every recursion are proportional to the number of players.

THEOREM 5.4. *Given a game $\Gamma = \langle I, \mathcal{A}, H', H, P, u \rangle$ and its root $h^0 = \emptyset$ as input, Algorithm 5 is polynomial with respect to the size of the game $|H|$.*

PROOF. Algorithm 5 is called recursively for every history of the game $h' \in H'$, where it is an intermediate history $h' \in H' \setminus H$ or an outcome $h \in H$. The number of operations to be performed at every recursion is $O(N)$, where N is the number of players. The total number of operations to be performed is thus $O(N \cdot |H'|)$. If the game is not degenerate, every player $i \in I$ can perform at least one action somewhere in the game, i.e., we suppose that every $S_i \neq \emptyset$. Therefore we have that $N < |H'|$ (usually $N \ll |H'|$). It is thus enough to prove that $|H'| = O(|H|)$. More specifically, for the intermediate histories $H' \setminus H$ of the game-tree it holds $|H' \setminus H| < |H|$ so that $|H'| < 2 \cdot |H|$. To show this let us define the function $f : H' \rightarrow \mathbb{N}$ such that $f(h') = \sum_{a \in \mathcal{A}(h')} f(h' + (a))$ if $h' \in H' \setminus H$ and $f(h') = 1$ if $h' \in H$. The function f counts for every history h' the outcomes of the subgame $\Gamma(h')$ and thus $f(h^0) = |H|$. Let us then define the function $g : H' \rightarrow \mathbb{N}$ such that $g(h') = 1 + \sum_{a \in \mathcal{A}(h')} g(h' + (a))$ if $h' \in H' \setminus H$ and $g(h') = 1$ if $h' \in H$. The function g counts the intermediate histories for every subgame and thus $g(h^0) = |H' \setminus H|$. We prove now that $f(h') > g(h')$ for every $h' \in H'$. For an outcome $h \in H$ it holds $f(h) = 1 > 0 = g(h)$. We proceed by induction. We suppose that for every other intermediate history $h' \in H' \setminus H$ and for every following node $h' + (a) \in \mathcal{A}$ it holds $f(h' + (a)) > g(h' + (a))$. By induction, we have that $f(h') = \sum_{a \in \mathcal{A}(h')} f(h' + (a)) = \sum_{a \in \mathcal{A}(h')} (f(h' + (a)) - 1) + |\mathcal{A}(h')| \geq \sum_{a \in \mathcal{A}(h')} g(h' + (a)) + |\mathcal{A}(h')| \geq g(h') + |\mathcal{A}(h')|$. Since the number of actions available is always $|\mathcal{A}(h')| > 1$ we have that $f(h') > g(h')$ for every $h' \in H'$. Therefore $|H| = f(h^0) > g(h^0) = |H' \setminus H|$, hence the proof. \square

6 CONCLUSIONS

In this paper we analysed Nash equilibria in extensive-form games with perfect information and perfect recall. The literature of the domain analyses the mathematical structure of the sets of strategies. Instead, we have focused on the set of possible outcomes decided by such strategies. We have observed that such sets can be deduced by comparing pairwise the outcomes of the game, we have introduced a new representation of extensive-form games as an undirected graph of their outcomes, which highlights a distinct property at the core of every couple of outcomes. This allowed us to introduce the first algorithm to determine whether an outcome is a realisation of a Nash equilibrium, generalising thus the results of [Zappalà et al., 2023] for two-player extensive-form games. Adopting such algorithm in parallel it is possible to determine all the realisations of Nash equilibria. However, we introduced another recursive algorithm that performs such enumeration in polynomial time with respect to the size of the game and thus generalises the celebrated backward induction algorithm.

We foresee the application of the new algorithms to the domain of games with more than two players, which has been almost unexplored in literature so far. The simplified notation of the graph form allows to better study those games whose outcomes belong to a space with a binary relation, e.g., timing games [Zappalà et al., 2022]. In the future we plan to study ways to characterise subgame perfect equilibria directly on the graph of the game; such result could provide new algorithms that are not based on the principle of backward induction and thus can be parallelized. Finally, a further open problem raised by this work is whether it is possible to characterise the set of outcomes of a strategy in an extensive-form game with imperfect information.

REFERENCES

- Charles Audet, Slim Belhaiza, and Pierre Hansen. 2009. A new sequence form approach for the enumeration and refinement of all extreme Nash equilibria for extensive form games. *International Game Theory Review* 11, 04 (2009), 437–451.
- Arthur Cayley. 1875. Mathematical questions with their solutions. *The Educational Times* 23 (1875), 18–19.
- Harold William Kuhn and Albert William Tucker. 1953. *Contributions to the Theory of Games*. Vol. 2. Princeton University Press.
- John F Nash Jr. 1950. Equilibrium points in n-person games. *Proceedings of the national academy of sciences* 36, 1 (1950), 48–49.

- Reinhard Selten. 1965. Spieltheoretische behandlung eines oligopolmodells mit nachfrageträgheit: Teil i: Bestimmung des dynamischen preisgleichgewichts. *Zeitschrift für die gesamte Staatswissenschaft/Journal of Institutional and Theoretical Economics* H. 2 (1965), 301–324.
- Jakub Szymanik. 2013. Backward Induction is PTIME-complete. In *International Workshop on Logic, Rationality and Interaction*. Springer, 352–356.
- Paolo Zappalà, Amal Benhamiche, Matthieu Chardy, Francesco De Pellegrini, and Rosa Figueiredo. 2023. Nouvelle modélisation des jeux extensifs basée sur des graphes. In *24ème congrès annuel de la Société Française de Recherche Opérationnelle et d’Aide à la Décision*. Rennes, France. <https://hal.science/hal-03953563>
- Paolo Zappalà, Amal Benhamiche, Matthieu Chardy, Francesco De Pellegrini, and Rosa Figueiredo. 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)*. 217–224. <https://doi.org/10.23919/WiOpt56218.2022.9930538>

A EXCLUDING CLIQUES

In this section we present the problems of the maximal excluding clique (cf. Algorithm 2) and of the excluding clique (cf. Algorithm 3) and the results obtained by [Zappalà et al., 2023].

Let us consider the graph $\langle H, E \rangle$ of Fig. 7 and a set $X \subset H$. The problem of the maximal excluding clique is to determine whether there is a set C that includes no elements of X and induces a maximal clique in the graph. The problem of the excluding clique is instead to determine whether there is a set C' such that for all the elements in X there is at least one in C' not connected to it. For instance, in the example of Fig. 7b the clique $C' = \{h_2, h_3\}$ is an excluding clique for $\{h_5, h_6, h_7, h_8, h_{10}\}$. Indeed, h_5 and h_8 are not connected to either $\{h_2, h_3\}$; h_6 is not connected to h_3 , while h_7 and h_8 are not connected to h_2 .

Given the partition $H = V \cup X$ and the graph $\langle V \cup X, E \rangle$ a linear formulation of the problem of the excluding clique with input $\langle V, X, E \rangle$ is:

$$\begin{aligned} x_i + x_{i'} &\leq 1 \quad i, i' \in V, (i, i') \notin E \\ \sum_{i:(i,j) \notin E} x_i &\geq 1 \quad \forall j \in X \\ x_i &\in \{0, 1\} \quad \forall i \in V. \end{aligned}$$

An excluding clique exists in the graph if and only if the above system provides a solution. The two following results hold [Zappalà et al., 2023].

THEOREM A.1. *The problem of the maximal excluding clique and the problem of the maximal clique are equivalent.*

THEOREM A.2. *In a generic graph the problem of the maximal clique is NP-complete.*

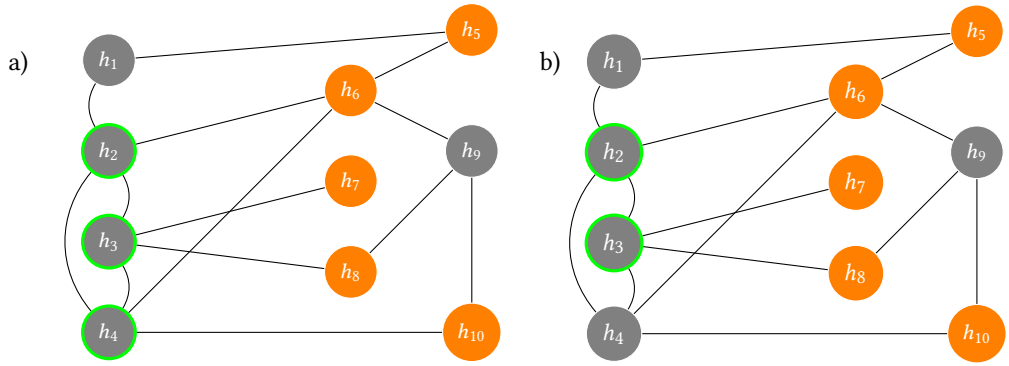


Fig. 7. **Equivalence of problems (MC) and (EC).** a) Let us consider problem (MC) with $H = \{h_1, h_2, h_3, h_4, h_5, h_6, h_7, h_8, h_9, h_{10}\}$ and $X = \{h_5, h_6, h_7, h_8, h_{10}\}$. A maximal clique that solves (MC) is $C = \{h_2, h_3, h_4\}$; b) Let us consider problem (EC) with $V = \{h_1, h_2, h_3, h_4\}$ and $X = \{h_5, h_6, h_7, h_8, h_{10}\}$. A clique that solves (EC) is $C' = \{h_2, h_3\}$.