Nash Equilibria in Games over Graphs Equipped with a Communication Mechanism

Patricia Bouyer

LSV, CNRS, ENS Paris-Saclay, Université Paris-Saclay, France

bouyer@lsv.fr

Nathan Thomasset

LSV, ENS Paris-Saclay, CNRS, Université Paris-Saclay, France

nathan.thomasset@ens-paris-saclay.fr

Abstract

We study pure Nash equilibria in infinite-duration games on graphs, with partial visibility of actions but communication (based on a graph) among the players. We show that a simple communication mechanism consisting in reporting the deviator when seeing it and propagating this information is sufficient for characterizing Nash equilibria. We propose an epistemic game construction, which conveniently records important information about the knowledge of the players. With this abstraction, we are able to characterize Nash equilibria which follow the simple communication pattern via winning strategies. We finally discuss the size of the construction, which would allow efficient algorithmic solutions to compute Nash equilibria in the original game.

2012 ACM Subject Classification Theory of computation; Theory of computation → Solution concepts in game theory; Theory of computation → Verification by model checking

Keywords and phrases Multiplayer games, Nash equilibria, partial information

Digital Object Identifier 10.4230/LIPIcs.MFCS.2019.9

Related Version Full version available at `http://arxiv.org/abs/1906.07753`

Funding This work has been partly supported by ERC project EQualIS (FP7-308087).

1 Introduction

Multiplayer concurrent games over graphs allow to model rich interactions between players. Those games are played as follows. In a state, each player chooses privately and independently an action, defining globally a move (one action per player); the next state of the game is then defined as the successor (on the graph) of the current state using that move; players continue playing from that new state, and form a(n infinite) play. Each player then gets a reward given by a payoff function (one function per player). In particular, objectives of the players may not be contradictory: those games are non-zero-sum games, contrary to two-player games used for controllers or reactive synthesis [15, 12].

Using solution concepts borrowed from game theory, one can describe interactions among the players, and in particular rational behaviours of selfish players. One of the most basic and classically studied solution concepts is that of Nash equilibria [13]. A Nash equilibrium is a strategy profile where no player can improve her payoff by unilaterally changing her strategy. The outcome of a Nash equilibrium can therefore be seen as a rational behaviour of the system. While very much studied by game theorists, e.g. over (repeated) matrix games, such a concept (and variants thereof) has been only rather recently studied over infinite-duration games on graphs. Probably the first works in that direction are [9, 8, 16, 17]. Several series of works have followed. To roughly give an idea of the existing results, pure Nash equilibria always exist in turn-based games for ω-regular objectives [19] but not in concurrent games, even with simple objectives; they can nevertheless be computed [19, 4, 7, 3] for large classes

44th International Symposium on Mathematical Foundations of Computer Science (MFCS 2019).
Editors: Peter Rossmanith, Pinar Heggernes, and Joost-Pieter Katoen; Article No. 9; pp. 9:1–9:14
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

of objectives. The problem becomes harder with mixed (that is, stochastic) Nash equilibria, for which we often cannot decide the existence [18, 5].

Computing Nash equilibria requires to (i) find a good behaviour of the system; (ii) detect deviations from that behaviour, and identify deviating players; (iii) punish them. This simple characterization of Nash equilibria is made explicit in [10]. Variants of Nash equilibria require slightly different ingredients, but they are mostly of a similar vein.

Many of those results are proven using the construction of a two-player game, in which winning strategies correspond (in some precise sense) to Nash equilibria in the original game. This two-player game basically records the knowledge of the various players about everything which can be uncertain: (a) the possible deviators in [4], and (b) the possible states the game can be in [3]. Extensions of this construction can be used for other solution concepts like robust equilibria [7] or rational synthesis [11].

In this work, we consider infinite-duration games on graphs, in which the game arena is perfectly known by all the players, but players have only a partial information on the actions played by the other players. The partial-information setting of this work is inspired by [14]: it considers repeated games played on matrices, where players only see actions of their neighbours. Neighbours are specified by a communication graph. To ensure a correct detection of deviators, the solution is to propagate the identity of the deviator along the communication graph. A fingerprint (finite sequence of actions) of every player is agreed at the beginning, and the propagation can be made properly if and only if the communication graph is 2-connected, ensuring large sets of Nash equilibria (formalized as a folk theorem). Fingerprints are not adapted to the setting of graphs, since they may delay the time at which a player will learn the identity of the deviator, which may be prohibitive if a bad component of a graph is then reached.

We therefore propose to add real communication among players. Similarly to [14], a player can communicate only with her neighbours (also specified by a communication graph), but can send arbitrary messages (modelled as arbitrary words over alphabet $\{0,1\}$). We assume that visited states are known by the players, hence only the deviator (if any) may be unknown to the players. In this setting, we show the following results:

- We show that a very simple epidemic-like communication mechanism is sufficient for defining Nash equilibria. It consists in (a) reporting the deviator (for the neighbours of the deviator) as soon as it is detected, and (b) propagating this information (for the other players).
- We build an epistemic game, which tracks those strategy profiles which follow the above simple communication pattern. This is a two-player turn-based game, in which `Eve` (the first player) suggest moves, and `Adam` (the second player) complies (to generate the main outcome), or not (to mimic single-player deviations). The correctness of the construction is formulated as follows: there is a Nash equilibrium in the original game of payoff p if and only if there is a strategy for `Eve` in the epistemic game which is winning for p.
- We analyze the complexity of this construction.

Note that we do not assume connectedness of the communication graph, hence the particular case of a graph with no edges allows to recover the setting of [4] while a complete graph allows to recover the settings of [19, 7].

In Section 2, we define our model and give an example to illustrate the role of the communication graph. In Section 3, we prove the simple communication pattern. In Section 4, we construct the epistemic game and discuss its correctness. In Section 5, we discuss complexity issues. All proofs are available in the technical report [6].

2 Definitions

We use the standard notations $\mathbb{R}$ (resp. $\mathbb{Q}$, $\mathbb{N}$) for the set of real (resp. rational, natural) numbers. If S is a subset of $\mathbb{R}$, we write $\overline{S}$ for $S \cup \{-\infty, +\infty\}$.

Let S be a finite set and $R \subseteq S$. If m is an S-vector over some set Σ, we write $m(R)$ (resp. $m(-R)$) for the vector composed of the R-components of m (resp. all but the R-components). We also use abusively the notations $m(i)$ (resp. $m(-i)$) when i is a single element of S, and may sometimes even use m_i if this is clear in the context. Also, if $s \in S$ and $a \in \Sigma$, then $m[s/a]$ is the vector where the value $m(s)$ is replaced by a.

If S is a finite set, we write S^* (resp. S^+, S^ω) for the set of words (resp. non-empty word, infinite words) defined on alphabet S.

2.1 Concurrent games and communication graphs

We use the model of concurrent multi-player games [4], based on the two-player model of [1].

▶ **Definition 1.** *A* concurrent multiplayer game *is a tuple* $\mathcal{G} = \langle V, v_{\mathsf{init}}, \mathsf{Act}, \mathscr{P}, \Sigma, \mathsf{Allow}, \mathsf{Tab}, (\mathsf{payoff}_a)_{a\in\mathscr{P}}\rangle$*, where* V *is a finite set of vertices,* $v_{\mathsf{init}} \in V$ *is the initial vertex,* Act *is a finite set of actions,* $\mathscr{P}$ *is a finite set of players,* Σ *is a finite alphabet,* $\mathsf{Allow}\colon V \times \mathscr{P} \to 2^{\mathsf{Act}} \setminus \{\emptyset\}$ *is a mapping indicating the actions available to a given player in a given vertex,*[1] $\mathsf{Tab}\colon V \times \mathsf{Act}^{\mathscr{P}} \to V$ *associates, with a given vertex and a given action tuple the target vertex, for every* $a \in \mathscr{P}$*,* $\mathsf{payoff}_a\colon V^\omega \to \mathbb{D}$ *is a payoff function with values in a domain* $\mathbb{D} \subseteq \overline{\mathbb{R}}$*.*

An element of $\mathsf{Act}^{\mathscr{P}}$ is called a *move*. Standardly (see [1] for two-player games and [4] for the multiplayer extension), concurrent games are played as follows: from a given vertex v, each player selects independently an action (allowed by Allow), which altogether form a move m; then, the game proceeds to the next vertex, given by $\mathsf{Tab}(v, m)$; the game continues from that new vertex.

Our setting will refine this model, in that at each round, each player will also broadcast a message, which will be received by some of the players. The players that can receive a message will be specified using a communication graph that we will introduce later. The role of the messages will remain unclear until we commit to the definition of a strategy.

Formally, a *full history* h in $\mathcal{G}$ is a finite sequence

$$v_0 \cdot (m_0, \mathsf{mes}_0) \cdot v_1 \cdot (m_1, \mathsf{mes}_1) \cdot v_2 \ldots (m_{s-1}, \mathsf{mes}_{s-1}) \cdot v_s \in V \cdot \left(\left(\mathsf{Act}^{\mathscr{P}} \times (\{0,1\}^*)^{\mathscr{P}}\right) \cdot V\right)^*$$

such that for every $0 \leq r < s$, for every $a \in \mathscr{P}$, $m_r(a) \in \mathsf{Allow}(v_r, a)$, and $v_{r+1} = \mathsf{Tab}(v_r, m_r)$. For every $0 \leq r < s$, for every $a \in \mathscr{P}$, the set $\mathsf{mes}_r(a)$ is the message appended to action $m_r(a)$ at step $r+1$, which will be broadcast to some other players. For readability we will also write h as

$$v_0 \xrightarrow{m_0, \mathsf{mes}_0} v_1 \xrightarrow{m_1, \mathsf{mes}_1} v_2 \ldots \xrightarrow{m_{s-1}, \mathsf{mes}_{s-1}} v_s$$

We write $\mathit{vertices}(h) = v_0 \cdot v_1 \cdots v_s$, and $\mathit{last}(h)$ for the last vertex of h (that is, v_s). If $r \leq s$, we also write $h_{\geq r}$ (resp. $h_{\leq r}$) for the suffix $v_r \cdot (m_r, \mathsf{mes}_r) \cdot v_{r+1} \cdot (m_{r+1}, \mathsf{mes}_{r+1}) \ldots (m_{s-1}, \mathsf{mes}_{s-1}) \cdot v_s$ (resp. prefix $v_0 \cdot (m_0, \mathsf{mes}_0) \cdot v_1 \cdot (m_1, \mathsf{mes}_1) \ldots (m_{r-1}, \mathsf{mes}_{r-1}) \cdot v_r$). We write $\mathsf{Hist}_{\mathcal{G}}(v_0)$ (or simply $\mathsf{Hist}(v_0)$ if $\mathcal{G}$ is clear in the context) for the set of full histories in $\mathcal{G}$ that start at v_0. If $h \in \mathsf{Hist}(v_0)$ and $h' \in \mathsf{Hist}(\mathit{last}(h))$, then we write $h \cdot h'$ for the obvious concatenation of histories (it then belongs to $\mathsf{Hist}(v_0)$).

[1] This condition ensures that the game is non-blocking.

We add a *communication (directed) graph* $G = (\mathscr{P}, E)$ to the context. The set of vertices of G is the set of players, and edges define a neighbourhood relation. An edge $(a, b) \in E$ (with $a \neq b$) means that player b can see which actions are played by player a together with the messages broadcast by player a. Later we write $a \rightarrowtriangle b$ whenever $(a, b) \in E$ or $a = b$, and $\mathbb{n}(b) = \{a \in \mathscr{P} \mid a \rightarrowtriangle b\}$ for the so-called *neighbourhood* of b (that is, the set of players about which player b has information). If $a, b \in \mathscr{P}$, we write $\mathsf{dist}_G(a, b)$ for the distance in G from a to b ($+\infty$ if there is no path from a to b).

Let $a \in \mathscr{P}$ be a player. The projection of h for a is denoted $\pi_a(h)$ and is defined by

$$v_0 \cdot (m_0(\mathbb{n}(a)), \mathsf{mes}_0(\mathbb{n}(a))) \cdot v_1 \cdot (m_1(\mathbb{n}(a)), \mathsf{mes}_1(\mathbb{n}(a))) \cdot v_2 \ldots$$
$$\ldots (m_{s-1}(\mathbb{n}(a)), \mathsf{mes}_{s-1}(\mathbb{n}(a))) \cdot v_s \in V \cdot \Big(\Big(\mathsf{Act}^{\mathbb{n}(a)} \times (\{0,1\}^*)^{\mathbb{n}(a)}\Big) \cdot V\Big)^*$$

This will be the information available to player a. In particular, messages broadcast by the players are part of this information. Note that we assume perfect recall, that is, while playing, player a will remember all her past knowledge, that is, all of $\pi_a(h)$ if h has been played so far. We define the *undistinguishability relation* $\sim_a$ as the equivalence relation over full histories induced by π_a: for two histories h and h', $h \sim_a h'$ iff $\pi_a(h) = \pi_a(h')$. While playing, if $h \sim_a h'$, a will not be able to know whether h or h' has been played. We write $\mathsf{Hist}_{\mathcal{G},a}(v_0)$ for the set of histories for player a (also called a-histories) from v_0.

We extend all the above notions to infinite sequences in a straightforward way and to the notion of *full play*. We write $\mathsf{Plays}_{\mathcal{G}}(v_0)$ (or simply $\mathsf{Plays}(v_0)$ if $\mathcal{G}$ is clear in the context) for the set of full plays in $\mathcal{G}$ that start at v_0.

A *strategy* for a player $a \in \mathscr{P}$ from vertex v_0 is a mapping $\sigma_a \colon \mathsf{Hist}_{\mathcal{G}}(v_0) \to \mathsf{Act} \times \{0,1\}^*$ such that for every history $h \in \mathsf{Hist}_{\mathcal{G}}(v_0)$, $\sigma_a(h)[1] \in \mathsf{Allow}(\mathit{last}(h), a)$, where the notation $\sigma_a(h)[1]$ denotes the first component of the pair $\sigma(h)$. The value $\sigma_a(h)[1]$ represents the action that player a will do after h, while $\sigma_a(h)[2]$ is the message that she will append to her action and broadcast to all players b such that $a \rightarrowtriangle b$. The strategy σ_a is said *G-compatible* if furthermore, for all histories $h, h' \in \mathsf{Hist}(v_0)$, $h \sim_a h'$ implies $\sigma_a(h) = \sigma_a(h')$. In that case, σ_a can equivalently be seen as a mapping $\mathsf{Hist}_{\mathcal{G},a}(v_0) \to \mathsf{Act} \times \{0,1\}^*$. An *outcome* of σ_a is a(n infinite) play $\rho = v_0 \cdot (m_0, \mathsf{mes}_0) \cdot v_1 \cdot (m_1, \mathsf{mes}_1) \ldots$ such that for every $r \geq 0$, $\sigma_a(\rho_{\leq r}) = (m_r(a), \mathsf{mes}_r(a))$. We write $\mathsf{out}(\sigma_a, v_0)$ for the set of outcomes of σ_a from v_0.

A *strategy profile* is a tuple $\sigma_{\mathscr{P}} = (\sigma_a)_{a \in \mathscr{P}}$, where, for every player $a \in \mathscr{P}$, σ_a is a strategy for player a. The strategy profile is said *G-compatible* whenever each σ_a is G-compatible. We write $\mathsf{out}(\sigma_{\mathscr{P}}, v_0)$ for the unique full play from v_0, which is an outcome of all strategies part of $\sigma_{\mathscr{P}}$.

When $\sigma_{\mathscr{P}}$ is a strategy profile and σ'_d a player-d G-compatible strategy, we write $\sigma_{\mathscr{P}}[d/\sigma'_d]$ for the profile where player d plays according to σ'_d, and each other player a ($\neq d$) plays according to σ_a. The strategy σ'_d is a *deviation* of player d, or a *d-deviation* w.r.t. $\sigma_{\mathscr{P}}$. Such a d-deviation is said *profitable* w.r.t. $\sigma_{\mathscr{P}}$ whenever $\mathsf{payoff}_d\Big(\mathit{vertices}(\mathsf{out}(\sigma_{\mathscr{P}}, v_0))\Big) < \mathsf{payoff}_d\Big(\mathit{vertices}(\mathsf{out}(\sigma_{\mathscr{P}}[d/\sigma'_d], v_0))\Big)$.

▶ **Definition 2.** *A* Nash equilibrium *from v_0 is a G-compatible strategy profile $\sigma_{\mathscr{P}}$ such that for every $d \in \mathscr{P}$, there is no profitable d-deviation w.r.t. $\sigma_{\mathscr{P}}$.*

In this definition, deviation σ'_d needs not really to be G-compatible, since the only meaningful part of σ'_d is along $\mathsf{out}(\sigma[d/\sigma'_d], v_0)$, where there are no $\sim_d$-equivalent histories: any deviation can be made G-compatible without affecting the profitability of the resulting outcome.

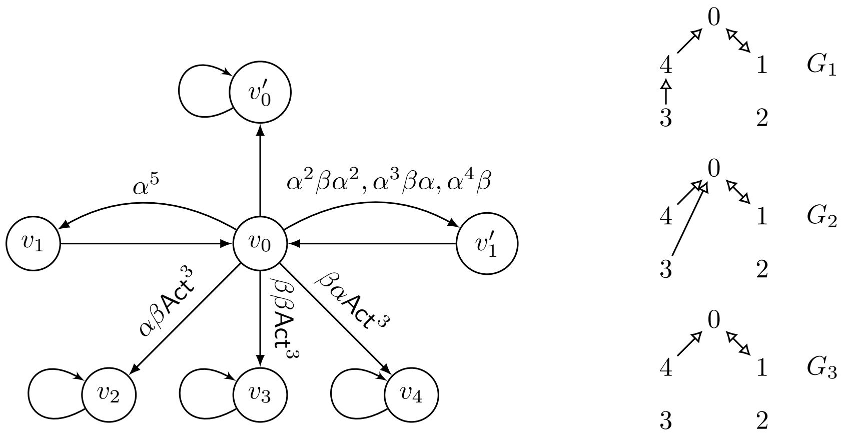

v'_0
α^5
$\alpha^2\beta\alpha^2, \alpha^3\beta\alpha, \alpha^4\beta$
v_1
v_0
v'_1
$\alpha\beta\mathsf{Act}^3$
$\beta\beta\mathsf{Act}^3$
$\beta\alpha\mathsf{Act}^3$
v_2
v_3
v_4
0
4
1
3
2
G_1
G_2
G_3

Figure 1 A five-player game (left) and three communication graphs (right); self-loops $a \rightarrow a$ omitted from the picture. The action alphabet is $\{\alpha, \beta\}$. The transition function is represented as arrows from one vertex to another labeled with the action profile(s) allowing to go from the origin vertex to the destination one. We write action profiles with length-5 words. Convention: no label means complementary labels (e.g. one goes from v_0 to v'_0 using any action profile that is not in $\alpha^5 + (\alpha\beta + \beta\beta + \beta\alpha)\mathsf{Act}^3 + \alpha^2(\beta\alpha^2 + \alpha\beta\alpha + \alpha^2\beta)$).

▶ Remark 3. Before pursuing our study, let us make clear what information players have: a player knows the full arena of the game and the whole communication graph; when playing the game, a player sees the states which are visited, and see actions of and messages from her neighbours (in the communication graph). When playing the profile of a Nash equilibrium, all players know all strategies, hence a player knows precisely what is expected to be the main outcome; in particular, when the play leaves the main outcome, each player knows that a deviation has occurred, even though she didn't see the deviator or received any message. Note that deviations which do not leave the main outcome may occur; in this case, only the neighbours of the deviator will know that such a deviation occurred; we will see that it is useless to take care of such deviations.

2.2 An example

We consider the five-player game described in Figure 1 in which we denote the players $\mathscr{P} = \{0, 1, 2, 3, 4\}$. The action alphabet is $\mathsf{Act} = \{\alpha, \beta\}$, and the initial vertex is assumed to be v_0. We suppose the payoff function vector is defined as (to be read as the list of payoffs of the players):

$$\mathsf{payoff}(\rho) = \begin{cases} (0,0,1,1,1) & \text{if } \rho \text{ visits } v_1 \text{ infinitely often} \\ (0,0,2,2,2) & \text{if } \rho \text{ visits } v_1 \text{ finitely often and } v'_1 \text{ infinitely often} \\ (0,0,0,2,2) & \text{if } \rho \text{ ends up in } v_2 \\ (0,0,2,0,2) & \text{if } \rho \text{ ends up in } v_3 \\ (0,0,2,2,0) & \text{if } \rho \text{ ends up in } v_4 \\ (0,0,3,3,3) & \text{if } \rho \text{ ends up in } v'_0 \end{cases}$$

We consider a (partial) strategy profile σ whose main outcome is:

$$\rho = \left(v_0 \cdot (\alpha^5, \mathsf{mes}_\epsilon) \cdot v_1 \cdot (\alpha^5, \mathsf{mes}_\epsilon)\right)^\omega$$

where $\mathsf{mes}_\epsilon(a) = \epsilon$ for every $a \in \mathscr{P}$. Note that players 0 and 1 cannot benefit from any deviation since their payoffs is uniformly 0. Then notice that no one alone can deviate from

ρ to v'_0. Now, the three players 2, 3 and 4 can alone deviate to v'_1 and (try to) do so infinitely often. We examine those deviations. If players 0 and 1 manage to learn who is the deviator, then, together, they can punish the deviator: if they learn that player 2 (resp. 3, 4) is the deviator, then they will enforce vertex v_2 (resp. v_3, v_4). If they do not manage to learn who is the deviator, then they will not know what to do, and therefore, in any completion of the strategy profile, there will be some profitable deviation for at least one of the players (hence there will not be any Nash equilibrium whose main outcome is ρ).

We examine now the three communication graphs G_1, G_2 and G_3 depicted in Figure 1. Using communication based on graph G_1, if player 4 deviates, then player 0 will see this immediately and will be able to communicate this fact to player 1; if player 3 deviates, then player 4 will see this immediately and will be able to communicate this fact to player 0, which will transmit to player 1; if player 2 deviates, then no one will see anything, hence they will deduce the identity of the deviator in all the cases.

Using communication based on graph G_2, if either player 3 or player 4 deviates, then player 0 will see this immediately and will be able to communicate this fact to player 1 using the richness of the communication scheme (words over $\{0, 1\}$). Like before, the identity of deviator 2 will be guessed after a while.

Using communication based on graph G_3, if player 4 deviates, then player 0 will see this immediately and will be able to communicate this fact to player 1 (as before); now, no one (except players 2 and 3) will be able to learn who is deviating, if player 2 or player 3 deviates.

We can conclude that there is a Nash equilibrium with graph G_1 or G_2 whose main outcome is ρ, but not with graph G_3.

2.3 Two-player turn-based game structures

Two-player turn-based game structures are specific cases of the previous model, where at each vertex, at most one player has more than one action in her set of allowed actions. But for convenience, we will give a simplified definition, with only objects that will be useful.

A two-player turn-based game structure is a tuple $G = \langle S, S_{\texttt{Eve}}, S_{\texttt{Adam}}, s_{\mathsf{init}}, A, \mathsf{Allow}, \mathsf{Tab}\rangle$, where $S = S_{\texttt{Eve}} \sqcup S_{\texttt{Adam}}$ is a finite set of states (states in $S_{\texttt{Eve}}$ belong to player Eve whereas states in $S_{\texttt{Adam}}$ belong to player Adam), $s_{\mathsf{init}} \in S$ is the initial state, A is a finite alphabet, $\mathsf{Allow}\colon S \to 2^A \setminus \{\emptyset\}$ gives the set of available actions, and $\mathsf{Tab}\colon S \times A \to S$ is the next-state function. If $s \in S_{\texttt{Eve}}$ (resp. $S_{\texttt{Adam}}$), $\mathsf{Allow}(s)$ is the set of actions allowed to Eve (resp. Adam) in state s.

In this context, strategies will use sequences of states. That is, if a denotes Eve or Adam, an a-strategy is a partial function $\sigma_a : S^* \cdot S_a \to A$ such that for every $H \in S^* \cdot S_a$ such that $\sigma_a(H)$ is defined, $\sigma_a(H) \in \mathsf{Allow}(\mathit{last}(H))$. Note that we do not include any winning condition or payoff function in the tuple, hence the name structure.

2.4 The problems we are looking at

We are interested in the constrained existence of a Nash equilibrium. For simplicity, we define rectangular threshold constraints, but could well impose more complex constraints, like Boolean combinations of linear constraints.

▶ **Problem 1** (Constrained existence problem). *Given a concurrent game $\mathcal{G} = \langle V, v_{\mathsf{init}}, \mathcal{P}, \mathsf{Act}, \Sigma, \mathsf{Allow}, \mathsf{Tab}, (\mathsf{payoff}_a)_{a\in\mathcal{P}}\rangle$, a communication graph G for $\mathcal{P}$, a predicate P over $\mathbb{R}^{|\mathcal{P}|}$, can we decide whether there exists a Nash equilibrium $\sigma_{\mathcal{P}}$ from v_{init} such that $\mathsf{payoff}(\mathit{vertices}(\mathsf{out}(\sigma_{\mathcal{P}}, v_{\mathsf{init}}))) \in P$? If so, compute one. If the predicate P is trivial, we simply speak of the existence problem.*

The case where the communication graph has no edge was studied in depth in [4], with a generic two-player construction called the suspect construction, allowing to decide the constrained existence problem for many kinds of payoff functions. The case where the communication graph is a clique was the subject of the work [7]. The general case of a communication graph has not been investigated so far, but induces interesting developments. In the next section, we show that we can restrict the search of Nash equilibria to the search of so-called normed strategy profiles, where the communication via messages follows a very simple pattern. We also argue that deviations which do not impact the visited vertices should not be considered in the analysis. Given those reductions, we then propose the construction of a two-player game, which will track those normed profiles. This construction is inspired by the suspect-game construction of [4] and of the epistemic game of [3].

3 Reduction to profiles following a simple communication mechanism

We fix a concurrent game $\mathcal{G} = \langle V, v_{\mathsf{init}}, \mathsf{Act}, \mathscr{P}, \Sigma, \mathsf{Allow}, \mathsf{Tab}, (\mathsf{payoff}_a)_{a\in\mathscr{P}}\rangle$ and a communication graph G. We assume that $v_{\mathsf{init}} = v_0$. We will reduce the search for Nash equilibria to the search for strategy profiles with a very specific shape. In particular, we will show that the richness of the communication offered by the setting is somehow useless, and that a very simple communication pattern will be sufficient for characterizing Nash equilibria.

In the following, we write mes_ϵ for the vector assigning the empty word ϵ to every player $a \in \mathscr{P}$. Furthermore, for every $d \in \mathscr{P}$, we pick some word $\mathsf{id}_d \in \{0,1\}^+$ which are all distinct (and different from ϵ).

We first define restrictions for deviations. Let $\sigma_{\mathscr{P}}$ be a strategy profile. A player-d deviation σ'_d is said *immediately visible* whenever, writing h for the longest common prefix of $\mathsf{out}(\sigma_{\mathscr{P}}, v_0)$ and $\mathsf{out}(\sigma_{\mathscr{P}}[d/\sigma'_d], v_0)$, $\mathsf{Tab}(\mathit{last}(h), m) \neq \mathsf{Tab}(\mathit{last}(h), m')$, where $m = \sigma_{\mathscr{P}}(h)[1]$ and $m' = \big(\sigma_{\mathscr{P}}[d/\sigma'_d](h)\big)[1]$ are the next moves according to $\sigma_{\mathscr{P}}$ and $\sigma_{\mathscr{P}}[d/\sigma'_d]$. That is, at the first position where player d changes her strategy, it becomes public information that a deviation has occurred (even though some players know who deviated – all the players a with $d \twoheadrightarrow a$ – , and some other don't know). It is furthermore called *honest* whenever for every $h' \in \mathsf{out}(\sigma_{\mathscr{P}}[d/\sigma'_d], v_0)$ such that h is a (non-strict) prefix of h', $\sigma'_d(h')[2] = \mathsf{id}_d$. Somehow, player d admits she deviated, and does so immediately and forever.

The simple communication mechanism that we will design consists in reporting the deviator (role of the direct neighbours of the deviator), and propagating this information along the communication graph (for all the other players). Formally, let $\sigma_{\mathscr{P}}$ be a strategy profile, and let ρ be its main outcome. The profile $\sigma_{\mathscr{P}}$ will be said *normed* whenever the following conditions hold:

1. for every $h \in \mathsf{out}(\sigma_{\mathscr{P}}) \cup \bigcup_{d\in\mathscr{P},\ \sigma'_d} \mathsf{out}(\sigma_{\mathscr{P}}[d/\sigma'_d], v_0)$, if $\mathit{vertices}(h)$ is a prefix of $\mathit{vertices}(\rho)$, then for every $a \in \mathscr{P}$, $\sigma_a(h)[2] = \epsilon$;
2. for every $d \in \mathscr{P}$, for every d-strategy σ'_d, if $h \cdot (m, \mathsf{mes}) \cdot v \in \mathsf{out}(\sigma_{\mathscr{P}}[d/\sigma'_d], v_0)$ is the first step out of $\mathit{vertices}(\rho)$, then for every $d \twoheadrightarrow a$, $\sigma_a(h \cdot (m, \mathsf{mes}) \cdot v)[2] = \mathsf{id}_d$;
3. for every $d \in \mathscr{P}$, for every d-strategy σ'_d, if $h \cdot (m, \mathsf{mes}) \cdot v \in \mathsf{out}(\sigma_{\mathscr{P}}[d/\sigma'_d], v_0)$ has left the main outcome for more than one step, then for every $a \in \mathscr{P}$, $\sigma_a(h \cdot (m, \mathsf{mes}) \cdot v)[2] = \epsilon$ if for all $b \twoheadrightarrow a$, $\mathsf{mes}(b) = \epsilon$ and $\sigma_a(h \cdot (m, \mathsf{mes}) \cdot v)[2] = \mathsf{id}_d$ if there is $b \twoheadrightarrow a$ such that $\mathsf{mes}(b) = \mathsf{id}_d$; note that this is well defined since at most one id can be transmitted.

The first condition says that, as long as a deviation is not visible, then no message needs to be sent; the second condition says that as soon as a deviation becomes visible, then messages denouncing the deviator should be sent by "those who know", that is, the (immediate) neighbours of the deviator; the third condition says that the name (actually, the id) of the deviator should propagate according to the communication graph in an epidemic way.

Note that the profiles discussed in Section 2.2 were actually normed.

▶ **Theorem 4.** *The existence of a Nash equilibrium $\sigma_{\mathcal{P}}$ with payoff p is equivalent to the existence of a normed strategy profile $\sigma'_{\mathcal{P}}$ with payoff p, which is resistant to immediately visible and honest single-player deviations.*

The proof of this theorem, which is rather technical, can be found in [6]. We only give some intuition here. First, we explain why being resistant to immediately visible and honest deviations is enough. Notice that as long as the sequence of vertices follows the main outcome, then one can simply ignore the deviation and act only when the deviation becomes visible, in a way as if the deviator had started deviating only at this moment. This will be enough to punish the deviator. The "honest" part comes from the fact that one should simply ignore the messages sent by the deviator as it can be only in her interest to not ignore them (if it was not, then why would she send any message at all?).

Second, we show why no one should communicate as long as the sequence of vertices follows the main outcome. The reason is that if no one has deviated then any message is essentially useless, and if a deviation has happened, as explained earlier it can just be ignored as long as it has not become visible.

Finally, we demonstrate why the richness of the communication mechanism is in a way useless. Intuitively, one can understand that the only factors that should matter when playing are the sequence of the vertices that have been visited (because payoff functions only take into account the visited vertices) and the identity of the deviator. Thus the messages should only be used so that players can know of the identity of the deviator in the fastest possible way, and we show that nothing is faster than a sort of epidemic mechanism where one simply broadcasts the identity of the deviator whenever one received the information.

4 The epistemic game abstraction

We fix a concurrent game $\mathcal{G} = \langle V, v_{\mathsf{init}}, \mathsf{Act}, \mathcal{P}, \Sigma, \mathsf{Allow}, \mathsf{Tab}, (\mathsf{payoff}_a)_{a\in\mathcal{P}}\rangle$ for the rest of the section, and G be a communication graph for $\mathcal{P}$. We will implement an epistemic abstraction, which will track normed strategy profiles, and check that there is no profitable immediately visible and honest single-player deviations.

4.1 Description of the epistemic game

A *situation* is a triple (d, I, K) in $\mathcal{P} \times 2^{\mathcal{P}} \times \left(2^{\mathcal{P}}\right)^{\mathcal{P}}$, which consists of a deviator $d \in \mathcal{P}$, a list of players I having received the information that d is the deviator, and a knowledge function K that associates to every player a a list of suspects $K(a)$; in particular, it should be the case that $d \in I$ and for every $a \in I$, $K(a) = \{d\}$. We write Sit for the set of situations.

The epistemic game $\mathcal{E}^G_{\mathcal{G}}$ of $\mathcal{G}$ and G is defined as a two-player game structure $\langle S, S_{\mathtt{Eve}}, S_{\mathtt{Adam}}, s_{\mathsf{init}}, \Sigma', \mathsf{Allow}', \mathsf{Tab}'\rangle$. We describe the states and the transitions leaving those states; in particular, components Σ', Allow', Tab' of the above tuple will only be implicitely defined.

$\mathtt{Eve}$'s states $S_{\mathtt{Eve}}$ consist of elements of $V \times 2^{\mathsf{Sit}}$ such that if (v, X) is a state then for all $a \in \mathcal{P}$ the set $\{(d, I, K) \in X \mid d = a\}$ is either a singleton or empty (there is at most one situation associated with a given player a). We write $\mathsf{dev}(X)$ the set $\{d \in \mathcal{P} \mid \exists (d, I, K) \in X\}$ of agents which are a deviator in one situation of X. If $d \in \mathsf{dev}(X)$, we write (d, I^X_d, K^X_d) for the unique triple belonging to X having deviator d. Hence, $X = \{(d, I^X_d, K^X_d) \mid d \in \mathsf{dev}(X)\}$.

Intuitively, an Eve's state (v, X) will correspond to a situation where the game has proceeded to vertex v, but, if $\mathsf{dev}(X) \neq \emptyset$, several players may have deviated. Each player $d \in \mathsf{dev}(X)$ may be responsible for the deviation; some people will have received a message denouncing d (those are in the set I_d^X), and some will deduce things from what they observe (this is given by K_d^X). Note that the (un)distinguishability relation of a player a will be deduced from X: if d deviated and $a \in I_d^X$, then a will know d deviated; if a is neither in I_d^X nor in $I_{d'}^X$, then a will not be able to know whether d or d' deviated (as we will prove later, in Lemma 5).

First let us consider the case where $X = \emptyset$, which is to be understood as the case where no deviation has arisen yet. In state $(v, \emptyset)$, Eve's actions are moves in $\mathcal{G}$ enabled in v. When she plays move $m \in \mathsf{Act}^{\mathscr{P}}$, the game progresses to Adam's state $((v, \emptyset), m) \in S_{\mathtt{Adam}}$ where Adam's actions are vertices $v' \in V$ such that there exists a player $d \in \mathscr{P}$ and an action $\delta \in \mathsf{Act}$ such that $\mathsf{Tab}(v, (m[d/\delta])) = v'$. When Adam plays v', either $v' = \mathsf{Tab}(v, m)$ and the game progresses to Eve's state $(v', \emptyset)$ or $v' \neq \mathsf{Tab}(v, m)$ and the game progresses to Eve's state (v', X') where:

- $d \in \mathsf{dev}(X')$ if and only if there is $\delta \in \mathsf{Act}$ such that $\mathsf{Tab}(v, (m[d/\delta])) = v'$. It means that given the next state v', d is a possible deviator;
- if $d \in \mathsf{dev}(X')$, then:
  - $I_d^{X'} = \{a \in \mathscr{P} \mid d \rightarrowtriangle a\}$;
  - for every $a \in I_d^{X'}$, $K_d^{X'}(a) = \{d\}$;
  - for every $a \notin I_d^{X'}$, $K_d^{X'}(a) = \{b \in \mathscr{P} \mid \exists \beta \in \mathsf{Act} \text{ s.t. } \mathsf{Tab}(v, (m[b/\beta])) = v'\} \setminus \{b \in \mathscr{P} \mid b \rightarrowtriangle a\}$. Those are all the players that can be suspected by a, given the vertex v', and the absence of messages so far.

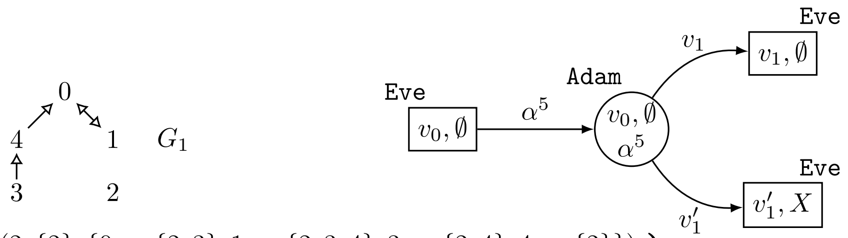
We write $X' = \mathsf{upd}((v, \emptyset), m, v')$. Note that $X' = \emptyset$ whenever (and only when) $\mathsf{Tab}(v, m) = v'$.

In a state $(v, X) \in S_{\mathtt{Eve}}$ where $X \neq \emptyset$, Eve's actions consist of functions from $\mathsf{dev}(X)$ to $\mathsf{Act}^{\mathscr{P}}$ that are compatible with players' knowledge, that is: $f : \mathsf{dev}(X) \to \mathsf{Act}^{\mathscr{P}}$ is an action enabled in (v, X) if and only if (i) for all $d \in \mathsf{dev}(X)$, for each $a \in \mathscr{P}$, $f(d)(a) \in \mathsf{Allow}(v, a)$, (ii) for all $d, d' \in \mathsf{dev}(X)$, for all $a \in \mathscr{P}$, if $a \notin I_d^X \cup I_{d'}^X$ and $K_d^X(a) = K_{d'}^X(a)$ then $f(d)(a) = f(d')(a)$;[2] that is, if a player has not received any message so far but has the same knowledge about the possible deviators in two situations, then Eve's suggestion for that player's action must be the same in both situations. When Eve plays action f in (v, X), the next state is $((v, X), f) \in S_{\mathtt{Adam}}$, where Adam's actions correspond to states of the game that are compatible with (v, X) and f, that is states v' such that there exists $d \in \mathsf{dev}(X)$ and $\delta \in \mathsf{Act}$ such that $\mathsf{Tab}(v, f(d)[d/\delta]) = v'$.

When Adam chooses the action v' in $((v, X), f)$, the game progresses to Eve's state (v', X'), where:

- $d \in \mathsf{dev}(X')$ if and only if $d \in \mathsf{dev}(X)$ and there exists $\delta \in \mathsf{Act}$ such that $\mathsf{Tab}(v, f(d)[d/\delta]) = v'$. It corresponds to a case where d was already a possible deviator and can continue deviating so that the game goes to v';
- if $d \in \mathsf{dev}(X')$, then:
  - $I_d^{X'} = I_d^X \cup \{a \in \mathscr{P} \mid \exists b \in I_d^X \text{ s.t. } b \rightarrowtriangle a\}$. New players receive a message with the deviator id;
  - for every $a \in I_d^{X'}$, $K_d^{X'}(a) = \{d\}$;

[2] Note in particular that "$K_d^X(a)$ singleton" does not imply $a \in I_d^X$, those are two distinguishable situations: the message with the identity of the deviator may not have been received in the first case, while it has been received in the second case.



$$X = \left\{\begin{array}{l}(2, \{2\}, \{0 \mapsto \{2,3\}, 1 \mapsto \{2,3,4\}, 3 \mapsto \{2,4\}, 4 \mapsto \{2\}\}),\\ (3, \{3,4\}, \{0 \mapsto \{2,3\}, 1 \mapsto \{2,3,4\}, 2 \mapsto \{3,4\}\}),\\ (4, \{0,4\}, \{1 \mapsto \{2,3,4\}, 2 \mapsto \{3,4\}, 3 \mapsto \{2,4\}\})\end{array}\right\}$$

Figure 2 Part of the epistemic game corresponding to the game described in Figure 1 (with graph G_1). This does not represent the whole epistemic game and a lot of actions accessible in the states we show here are not written. In situations (d, I, K) we describe K by the list of its values for players $a \notin I$, as for all a in I we have $K(a) = \{d\}$ by definition.

- for every $a \notin I_d^{X'}$, $K_d^{X'}(a) = \{b \in K_d^X(a) \mid \exists \beta \in \mathsf{Act} \text{ s.t. } \mathsf{Tab}(v, f(b)[b/\beta]) = v'\} \setminus \{c \in \mathcal{P} \mid \mathsf{dist}_G(c, a) \leq \max\{\mathsf{dist}_G(c, c') \mid c' \in I_c^X\} + 1\}$. Those are the players that could have deviated but for which player a would not have received the signal yet.

We write $X' = \mathsf{upd}((v, X), f, v')$. Note that $X' \neq \emptyset$ and that $\mathsf{dev}(X') \subseteq \mathsf{dev}(X)$.

We let $R = (v_0, X_0) \cdot ((v_0, X_0), f_0) \cdot (v_1, X_1) \ldots$ be an infinite play from $(v_0, X_0) = (v_{\mathsf{init}}, \emptyset)$. We write *visited*$(R)$ for $v_0 v_1 \cdots \in V^\omega$ the sequence of vertices visited along R. We also define $\mathsf{dev}(R) = \emptyset$ if $X_r = \emptyset$ for every r, and $\mathsf{dev}(R) = \lim_{r \to +\infty} \mathsf{dev}(X_r)$ otherwise. This is the set of possible deviators along R.

4.1.1 Winning condition of Eve

A zero-sum game will be played on the game structure $\mathcal{E}_\mathcal{G}^G$, and the winning condition of Eve will be given on the branching structure of the set of outcomes of a strategy for Eve, and not individually on each outcome, as standardly in two-player zero-sum games. We write $s_{\mathsf{init}} = (v_{\mathsf{init}}, \emptyset)$ for the initial state. Let $p = (p_a)_{a \in \mathcal{P}} \in \mathbb{R}^\mathcal{P}$, and ζ be a strategy for Eve in $\mathcal{E}_\mathcal{G}^G$; it is said winning for p from s_{init} whenever $\mathsf{payoff}(\mathit{visited}(R)) = p$, where R is the unique outcome of ζ from s_{init} where Adam complies to Eve's suggestions, and for every other outcome R' of ζ, for every $d \in \mathsf{dev}(R')$, $\mathsf{payoff}_d(\mathit{visited}(R')) \leq p_d$.

4.2 An example

In Figure 2 we present a part of the epistemic game corresponding to the game we described in Figure 1 with graph G_1. In state $(v_0, \emptyset)$, Eve can play the action profile α^5 and make the game go to $((v_0, \emptyset), \alpha^5)$ where Adam can either play $v_1 = \mathsf{Tab}(v_0, \alpha^5)$ (we say that Adam complies with Eve) or choose a different state accessible from v_0 and an action profile that consists in a single-player deviation from α, for instance $v'_1 = \mathsf{Tab}(v_0, \alpha^2\beta\alpha^2) = \mathsf{Tab}(v_0, \alpha^3\beta\alpha) = \mathsf{Tab}(v_0, \alpha^4\beta)$. If Adam chooses v'_1, then three players are possible deviators: 2, 3 and 4. We write X for the corresponding set of situations, and we already know that $\mathsf{dev}(X) = \{2, 3, 4\}$.

- If player 2 is the deviator, then no one (except himself) directly receives this information. Player 0 knows that player 4 did not deviate (since 4→0 in G_1), hence $K_2^X(0) = \{2, 3\}$; Player 1 has no information hence $K_2^X(1) = \{2, 3, 4\}$; Player 3 knows that he is not the deviator but cannot know more, hence $K_2^X(3) = \{2, 4\}$; Finally, player 4 can deduce many things: he knows he is not the deviator, and he saw that player 3 is not the deviator (since 3→4 in G_1), hence $K_2^X(4) = \{2\}$.

- If player 3 is the deviator, then both players 3 and 4 get the information, hence $I_3^X = \{3, 4\}$. Other players can guess some things, for instance player 0 sees that player 4 cannot be the deviator, this is why $K_3^X(0) = \{2, 3\}$. Etc.
- The reasoning for player 4 is similar.

In the situation we have just described, when the game will proceed to v_1', then either player 0 knows that player 4 has deviated, or he knows that player 4 didn't deviate but he suspects both 2 and 3. On the other hand, player 4 will precisely know who deviated. And player 3 knows whether he deviated or not, but if he didn't, then he cannot know whether it was player 2 or player 4 who deviated. This knowledge is stored in situation X we have described, and which is fully given in Figure 2.

Let us now illustrate how actions of Eve are defined in states with a non-empty set of situations. Assume we are in Eve's state (v_0, X), with X as previously defined. From that state, an action for Eve is a mapping $f : \{2, 3, 4\} \to \mathsf{Act}^{\mathscr{P}}$ such that:

$$f(2)(0) = f(3)(0) \quad f(2)(1) = f(3)(1) = f(4)(1) \quad f(3)(2) = f(4)(2) \quad f(2)(3) = f(4)(3)$$

The intuition behind these constraints is the following: Player 0 knows whether Player 4 deviated or not, but in the case she did not cannot know whether Player 2 or Player 3 deviated; Player 1 does not know who deviated, hence should play the same action in the three cases (that she cannot distinguish); Player 2 does only know whether she deviated hence in the case she did not cannot know whether Player 3 or Player 4 deviated; the case for Player 3 is similar; finally Player 4 knows for sure who deviated: she saw if Player 3 deviated and knows whether she herself deviated, thus can distinguish between the three cases.

4.3 Correctness of the epistemic game construction

When constructing the epistemic game, we mentioned that Eve's states will allow to properly define the undistinguishability relation for all the players. Towards that goal, we show by an immediate induction the following result:

▶ **Lemma 5.** *If (v, X) is an Eve's state reachable from some $(v_0, \emptyset)$ in $\mathcal{E}_{\mathcal{G}}^G$, then for all $d \in \mathsf{dev}(X)$:*

- *for all $a \in I_d^X$, $K_d^X(a) = \{d\}$;*
- *for all $a \notin I_d^X$, $K_d^X(a) = \mathsf{dev}(X) \setminus \{d' \in \mathsf{dev}(X) \mid a \in I_{d'}^X\}$.*

In particular, for all $d, d' \in \mathsf{dev}(X)$, for all $a \notin I_d^X \cup I_{d'}^X$, $K_d^X(a) = K_{d'}^X(a)$.

So, either a player a will have received from a neighbour the identity of the deviator, or she will not have received any deviator identity yet, and she will have a set of suspected deviators that she will not be able to distinguish.

This allows to deduce the following correspondence between $\mathcal{G}$ and $\mathcal{E}_{\mathcal{G}}^G$:

▶ **Proposition 6.** *There is a winning strategy for Eve in $\mathcal{E}_{\mathcal{G}}^G$ for payoff p if and only if there is a normed strategy profile in $\mathcal{G}$, whose main outcome has payoff p and which is resistant to single-player immediately visible and honest deviations.*

The proof of correctness of the epistemic game then goes through the following steps, which are detailed in [6]. First, given an Eve's strategy ζ, we build a function E_ζ associating with a-histories (for every $a \in \mathscr{P}$) in the original game Eve's histories in the epistemic game such that Eve plays according to ζ along E_ζ.

Then we use this function to create a strategy profile $\Omega(\zeta)$ for the original game where the action prescribed by this profile to player a after history h corresponds in some sense to

$\zeta(E_\zeta(h))(d)(a)$, where d is a suspected deviator according to player a. This works because, thanks to Lemma 5, we know that either player a knows who the deviator is, or player a has a subset of suspect deviators and Eve's suggestion for a (by construction of $\mathcal{E}_\mathcal{G}^G$) is the same for all those possible deviators.

Finally we prove that if ζ is a winning strategy for Eve then $\Omega(\zeta)$ is both normed and resistant to single-player immediately visible and honest deviations in $\mathcal{G}$.

To prove the converse proposition we build a function Λ associating with Eve's histories in the epistemic game families of single-player histories in the original game. We then use this correspondence to build a function Υ associating with normed strategy profiles Eve's strategies in a natural way.

Finally we prove that if σ is normed and resistant to single-player immediately visible honest deviations, then $\Upsilon(\sigma)$ is a winning strategy for Eve.

Gathering results of Theorem 4 and of this proposition, we get the following theorem:

▶ **Theorem 7.** *There is a Nash equilibrium with payoff p in $\mathcal{G}$ if and only if there is a winning strategy for* Eve *in $\mathcal{E}_\mathcal{G}^G$ for payoff p.*

▶ Remark 8. Note that all the results are constructive, hence if one can synthesize a winning strategy for Eve in $\mathcal{E}_\mathcal{G}^G$, then one can synthesize a correponding Nash equilibrium in $\mathcal{G}$.

5 Complexity analysis

We borrow all notations of previous sections. A rough analysis of the size of the epistemic game $\mathcal{E}_\mathcal{G}^G$ gives an exponential bound. We will give a more precise bound, pinpointing the part with an exponential blowup. We write $\mathsf{diam}(G)$ for the diameter of G, that is $\mathsf{diam}(G) = \max\{\mathsf{dist}_G(a,b) \mid \mathsf{dist}_G(a,b) < +\infty\}$.

▶ **Lemma 9.** *Assuming that* Tab *is given explicitely in $\mathcal{G}$, the number of states in the reachable part of $\mathcal{E}_\mathcal{G}^G$ from $s_{\mathsf{init}} = (v_{\mathsf{init}}, \emptyset)$ is bounded by*

$$|S_{\mathtt{Eve}}| \le |V| + |V| \cdot |\mathsf{Tab}|^2 \cdot (\mathit{diam}(G) + 2) \qquad \text{and} \qquad |S_{\mathtt{Adam}}| \le |S_{\mathtt{Eve}}| \cdot |\mathsf{Act}|^{|\mathcal{P}|^2}$$

The number of edges is bounded by $|S_{\mathtt{Adam}}| + |S_{\mathtt{Adam}}| \cdot |S_{\mathtt{Eve}}|$.

If $|\mathcal{P}|$ is assumed to be a constant of the problem, then the size of $\mathcal{E}_\mathcal{G}^G$ is polynomial in the size of $\mathcal{G}$.

We will not detail algorithmics issues, but the winning condition of Eve in $\mathcal{E}_\mathcal{G}^G$ is very similar to the winning condition of Eve in the suspect-game construction of [4] (for Boolean or ordered objectives), or in the deviator-game construction of [7] (for mean-payoff), or in a closer context to the epistemic-game construction of [3]. Hence, when the size of the epistemic game is polynomial, rather efficient algorithms can be designed to compute Nash equilibria. For instance, in a setting where the size of $\mathcal{E}_\mathcal{G}^G$ is polynomial, using a bottom-up labelling algorithm similar to that of [2, Sect. 4.3], one obtains a polynomial space algorithm for deciding the (constrained) existence of a Nash equilibrium when payoffs are Boolean payoffs corresponding to parity conditions.

6 Conclusion

In this paper, we have studied multiplayer infinite-duration games over graphs, and focused on games where players can communicate with neighbours, given by a directed graph. We have shown that a very simple communication mechanism was sufficient to describe Nash equilibria.

This mechanism is sort of epidemic, in that if a player deviates, then his neighbours will see it and transmit the information to their own neighbours; the information then propagates along the communication graph. This framework encompasses two standard existing frameworks, one where the actions are invisible (represented with a graph with no edges), and one where all actions are visible (represented by the complete graph). We know from previous works that in both frameworks, one can compute Nash equilibria for many kinds of payoff functions. In this paper, we also show that we can compute Nash equilibria in this generalized framework, by providing a reduction to a two-player game, the so-called epistemic game construction. Winning condition in this two-player game is very similar to winning conditions encountered in the past, yielding algorithmic solution to the computation of Nash equilibria. We have also analyzed the size of the abstraction, which is polynomial when the number of players is considered as a constant of the problem.

The current framework assumes messages can be appended to actions by players, allowing a rich communication between players. The original framework of [14] did not allow additional messages, but did encode identities of deviators by sequences of actions. This was possible in [14] since games were repeated matrix games, but it is harder to see how we could extend this approach and how we could encode identities of players with actions, taking into account the graph structure. For instance, due to the graph, having too long identifiers might be prohibitive to transmit in a short delay the identity of the deviator. Nevertheless, that could be interesting to see if something can be done in this framework.

References

1 Rajeev Alur, Thomas A. Henzinger, and Orna Kupferman. Alternating-time Temporal Logic. *Journal of the ACM*, 49:672–713, 2002. doi:10.1145/585265.585270.

2 Patricia Bouyer. Games on graphs with a public signal monitoring. Research report, arXiv, 2017. arXiv:1710.07163.

3 Patricia Bouyer. Games on graphs with a public signal monitoring. In *Proc. 21st International Conference on Foundations of Software Science and Computation Structures (FoSSaCS'18)*, volume 10803 of *Lecture Notes in Computer Science*, pages 530–547. Springer, 2018.

4 Patricia Bouyer, Romain Brenguier, Nicolas Markey, and Michael Ummels. Pure Nash Equilibria in Concurrent Games. *Logical Methods in Computer Science*, 11(2:9), 2015.

5 Patricia Bouyer, Nicolas Markey, and Daniel Stan. Mixed Nash Equilibria in Concurrent Games. In *Proc. 33rd Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'14)*, volume 29 of *LIPIcs*, pages 351–363. Leibniz-Zentrum für Informatik, 2014.

6 Patricia Bouyer and Nathan Thomasset. Nash equilibria in games over graphs equipped with a communication mechanism. Research report, arXiv, 2019. arXiv:1906.07753.

7 Romain Brenguier. Robust Equilibria in Mean-Payoff Games. In *Proc. 19th International Conference on Foundations of Software Science and Computation Structures (FoSSaCS'16)*, volume 9634 of *Lecture Notes in Computer Science*, pages 217–233. Springer, 2016.

8 Krishnendu Chatterjee, Thomas A. Henzinger, and Marcin Jurdziński. Games with Secure Equilibria. *Theoretical Computer Science*, 365(1-2):67–82, 2006.

9 Krishnendu Chatterjee, Rupak Majumdar, and Marcin Jurdziński. On Nash Equilibria in Stochastic Games. In *Proc. 18th International Workshop on Computer Science Logic (CSL'04)*, volume 3210 of *Lecture Notes in Computer Science*, pages 26–40. Springer, 2004.

10 Rodica Condurache, Emmanuel Filiot, Raffaella Gentilini, and Jean-François Raskin. The Complexity of Rational Synthesis. In *Proc. 43rd International Colloquium on Automata, Languages and Programming (ICALP'16)*, volume 55 of *LIPIcs*, pages 121:1–121:15. Leibniz-Zentrum für Informatik, 2016.

11 Rodica Condurache, Youssouf Oualhadj, and Nicolas Troquard. The Complexity of Rational Synthesis for Concurrent Games. In *Proc. 29th International Conference on Concurrency Theory (CONCUR'18)*, volume 118 of *LIPIcs*, pages 38:1–38:15. Leibniz-Zentrum für Informatik, 2018.

12 Thomas A. Henzinger. Games in system design and verification. In *Proc. 10th Conference on Theoretical Aspects of Rationality and Knowledge (TARK'05)*, pages 1–4, 2005.

13 John F. Nash. Equilibrium Points in n-Person Games. *Proceedings of the National Academy of Sciences of the United States of America*, 36(1):48–49, 1950.

14 Jérôme Renault and Tristant Tomala. Repeated Proximity Games. *International Journal of Game Theory*, 27(4):539–559, 1998.

15 Wolfgang Thomas. Infinite Games and Verification. In *Proc. 14th International Conference on Computer Aided Verification (CAV'02)*, volume 2404 of *Lecture Notes in Computer Science*, pages 58–64. Springer, 2002. Invited Tutorial.

16 Michael Ummels. Rational Behaviour and Strategy Construction in Infinite Multiplayer Games. In *Proc. 26th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'06)*, volume 4337 of *Lecture Notes in Computer Science*, pages 212–223. Springer, 2006.

17 Michael Ummels. The Complexity of Nash Equilibria in Infinite Multiplayer Games. In *Proc. 11th International Conference on Foundations of Software Science and Computation Structures (FoSSaCS'08)*, volume 4962 of *Lecture Notes in Computer Science*, pages 20–34. Springer, 2008.

18 Michael Ummels and Dominik Wojtczak. The Complexity of Nash Equilibria in Limit-Average Games. In *Proc. 22nd International Conference on Concurrency Theory (CONCUR'11)*, volume 6901 of *Lecture Notes in Computer Science*, pages 482–496. Springer, 2011.

19 Michael Ummels and Dominik Wojtczak. The Complexity of Nash Equilibria in Stochastic Multiplayer Games. *Logical Methods in Computer Science*, 7(3), 2011.