Nash Equilibria for Reachability Objectives in Multi-player Timed Games[*]

Patricia Bouyer, Romain Brenguier, and Nicolas Markey

LSV, ENS Cachan & CNRS, France
{bouyer,brenguie,markey}@lsv.ens-cachan.fr

Abstract. We propose a procedure for computing Nash equilibria in multi-player timed games with reachability objectives. Our procedure is based on the construction of a finite concurrent game, and on a generic characterization of Nash equilibria in (possibly infinite) concurrent games. Along the way, we use our characterization to compute Nash equilibria in finite concurrent games.

1 Introduction

Timed games. Game theory (especially games played on graphs) has been used in computer science as a powerful framework for modelling interactions in embedded systems [15, 10]. Over the last fifteen years, games have been extended with the ability to depend on timing informations. Timed games allows for a more faithful representation of reactive systems, while preserving decidability of several important properties, such as the existence of a winning strategy for one of the agents to achieve her goal, whatever the other agents do [3]. Efficient algorithms exist and have been implemented, *e.g.* in the tool Uppaal Tiga [4].

Zero-sum vs. non-zero-sum games. In this purely antagonist view, games can be seen as two-player games, where one agent plays against another one. Moreover, the objectives of those two agents are opposite: the aim of the second player is simply to prevent the first player from winning her own objective. More generally, a (positive or negative) *payoff* can be associated with each outcome of the game, which can be seen as the amount the second player will have to pay to the first player. Those games are said to be *zero-sum*.

In many cases, however, games can be *non-zero-sum*, especially when they involve more than two agents, whose objectives may not be complementary. Such games appear *e.g.* in various problems in telecommunications, where several agents try to send data on a network [9]. Focusing only on surely-winning strategies in this setting may then be too narrow: surely-winning strategies must be winning against any behaviour of the other agents, and does not consider the fact that the other agents also try to achieve their own objectives.

Nash equilibria. In the non-zero-sum game setting, it is then more interesting to look for *equilibria*. For instance, a Nash equilibrium is a behaviour of the

[*] This work is partly supported by projects DOTS (ANR-06-SETI-003), QUASIMODO (FP7-ICT-STREP-214755) and GASICS (ESF-EUROCORES LogiCCC).

agents in which they play rationally, in the sense that no agent can get a better payoff if she, alone, modifies her strategy [13]. This corresponds to stable states of the game. Notice that a Nash equilibrium need not exist in general, and is not necessarily optimal: several equilibria can coexist, possibly with very different payoffs.

Our contribution. We extend the study of Nash equilibria for reachability objectives (where the payoff is 1 when the objective is reached, and 0 otherwise) in the setting of timed games, as defined in [8] (but extended to n players in the obvious way). Since timed games are non-deterministic, we introduce the notion of *pseudo-Nash equilibrium*, in which non-determinism is solved "optimally" (*i.e.*, only the best outcome is considered). This corresponds to letting the players "optimally" solve non-determinism, in such a way that they have no incentive to change their choice.

As is usual in the timed setting, we rely on a region-based abstraction, which in our context is a finite concurrent game. In order to prove that the abstraction preserves Nash equilibria, we define a characterization of Nash equilibria in (possibly infinite-state) concurrent games. This characterization is built on the new concept of *repellor sets*: the repellor set for a subset A of agents is, roughly, the set of states from which players in A will not be winning in any Nash equilibrium. We explain how to compute those sets, and how they can be used to characterize Nash equilibria.

We also use repellor sets to effectively compute Nash equilibria in finite games, which solves open problems in the setting of equilibria in finite games and gives a complete solution to our original problem.

Related work. Nash equilibria (and other related solution concepts such as subgame-perfect equilibria, secure equilibria, ...) have recently been studied in the setting of (untimed) games played on a graph [6, 7, 14, 16–19]. Most of them, however, focus on turn-based games. In the setting of concurrent games, mixed strategies (*i.e.*, strategies involving probabilistic choices) are arguably more relevant than pure (*i.e.*, non-randomized) strategies. However, adding probabilities to timed strategies (over both delays and actions) involves several important technical issues (even in zero-sum non-probabilistic timed games), and we defer the study of mixed-strategy Nash equilibria in timed games to future works.

For lack of space, only sketches of proofs are given in the paper. Full proofs can be found in [5].

2 Preliminaries

We begin with defining concurrent games and Nash equilibria.

2.1 Concurrent Games

A *transition system* is a 2-tuple $\mathcal{S} = \langle \mathsf{States}, \mathsf{Edg} \rangle$ where States is a (possibly uncountable) set of states and $\mathsf{Edg} \subseteq \mathsf{States} \times \mathsf{States}$ is the set of transitions.

Given a transition system $\mathcal{S}$, a *path* π in $\mathcal{S}$ is a non-empty sequence $(s_i)_{0 \le i < n}$ (where $n \in \mathbb{N} \cup \{+\infty\}$) of states of $\mathcal{S}$ such that $(s_i, s_{i+1}) \in \mathsf{Edg}$ for all $i < n - 1$. The *length* of π, denoted by $|\pi|$, is $n - 1$. The set of finite paths (also called *histories* in the sequel) of $\mathcal{S}$ is denoted by[1] $\mathsf{Hist}_{\mathcal{S}}$, the set of infinite paths (also called *plays*) of $\mathcal{S}$ is denoted by $\mathsf{Play}_{\mathcal{S}}$, and $\mathsf{Path}_{\mathcal{S}} = \mathsf{Hist}_{\mathcal{S}} \cup \mathsf{Play}_{\mathcal{S}}$ is the set of paths of $\mathcal{S}$. Given a path $\pi = (s_i)_{0 \le i < n}$ and an integer $j < n$, the *j-th prefix* of π, denoted by $\pi_{\le j}$, is the finite path $(s_i)_{0 \le i < j+1}$. If $\pi = (s_i)_{0 \le i < n}$ is a history, we write $\mathsf{last}(\pi) = s_{|\pi|}$.

We extend the definition of concurrent games given *e.g.* in [2] with non-determinism:

Definition 1. *A* non-deterministic concurrent game *is a* 7*-tuple* $\mathcal{G} = \langle \mathsf{States}, \mathsf{Edg}, \mathsf{Agt}, \mathsf{Act}, \mathsf{Mov}, \mathsf{Tab}, \preccurlyeq \rangle$ *in which:*

- $\langle \mathsf{States}, \mathsf{Edg} \rangle$ *is a transition system;*
- Agt *is a finite set of players (or agents);*
- Act *is a (possibly uncountable) set of actions;*
- $\mathsf{Mov}\colon \mathsf{States} \times \mathsf{Agt} \to 2^{\mathsf{Act}} \smallsetminus \{\varnothing\}$ *is a mapping indicating the actions available to a given player in a given state;*
- $\mathsf{Tab}\colon \mathsf{States} \times \mathsf{Act}^{\mathsf{Agt}} \to 2^{\mathsf{Edg}} \smallsetminus \{\varnothing\}$ *associates, in a given state, a set of moves of the players with the resulting set of transitions. It is required that if* $(s', s'') \in \mathsf{Tab}(s, (m_A)_{A \in \mathsf{Agt}})$, *then* $s' = s$.
- $\preccurlyeq\colon \mathsf{Agt} \to 2^{\mathsf{States}^\omega \times \mathsf{States}^\omega}$ *defines, for each player, a quasi-ordering on the runs of* $\mathcal{G}$*, called* preference relation*. We simply write* $\preccurlyeq_A$ *for* $\preccurlyeq (A)$.

In the rest of this paper, we restrict to simple qualitative preference relations given by reachability conditions for each player. Formally, we assume that the preference relation is given as a tuple $(\Omega_A)_{A \in \mathsf{Agt}}$ of sets of states, and is defined as follows: if a path π visits Ω_A, then we let $v_A(\pi) = 1$, otherwise $v_A(\pi) = 0$; we then say that path π' is preferred by Player A over path π, which is written $\pi \preccurlyeq_A \pi'$, whenever either $\pi = \pi'$ or $v_A(\pi) < v_A(\pi')$.

A *deterministic* concurrent game is a concurrent game where $\mathsf{Tab}(s, (m_A)_{A \in \mathsf{Agt}})$ is a singleton for every $s \in \mathsf{States}$ and $(m_A)_{A \in \mathsf{Agt}}$ with $m_A \in \mathsf{Mov}(s, A)$. A *turn-based* game is a concurrent game for which there exists a mapping $\mathsf{Owner}\colon \mathsf{States} \to \mathsf{Agt}$ such that, for every state $s \in \mathsf{States}$, the set $\mathsf{Mov}(s, A)$ is a singleton unless $A = \mathsf{Owner}(s)$.

In a non-deterministic concurrent game, from some state s, each player A selects one action m_A among its set $\mathsf{Mov}(s, A)$ of allowed actions (the resulting tuple $(m_A)_{A \in \mathsf{Agt}}$, which we may also write m_{Agt} in the sequel, is called a *move*). This corresponds to a set of transitions $\mathsf{Tab}(s, (m_A)_{A \in \mathsf{Agt}})$, one of which is applied and gives the next state of the game. In the sequel, we abusively write $\mathsf{Hist}_{\mathcal{G}}$, $\mathsf{Play}_{\mathcal{G}}$ and $\mathsf{Path}_{\mathcal{G}}$ for the corresponding set of paths in the underlying transition system of $\mathcal{G}$. We also write $\mathsf{Hist}_{\mathcal{G}}(s)$, $\mathsf{Play}_{\mathcal{G}}(s)$ and $\mathsf{Path}_{\mathcal{G}}(s)$ for the respective subsets of paths starting in state s.

[1] For this and the coming definitions, we indicate the underlying transition system as a subscript. This may be omitted in the sequel if no ambiguity may arise.

Definition 2. *Let $\mathcal{G}$ be a concurrent game, and $A \in \mathsf{Agt}$. A* strategy *for A is a mapping $\sigma_A\colon \mathsf{Hist} \to \mathsf{Act}$ such that for any $\pi \in \mathsf{Hist}$ it holds $\sigma_A(\pi) \in \mathsf{Mov}(\mathsf{last}(\pi), A)$.*

Given a coalition (i.e., a subset of agents) $P \subseteq \mathsf{Agt}$, a strategy σ_P for coalition P is a tuple of strategies, one for each player in P. We write $\sigma_P = (\sigma_A)_{A\in P}$ for such a strategy. A strategy profile *is a strategy for the coalition Agt. We write $\mathsf{Strat}^A_{\mathcal{G}}$ for the set of strategies of player A in $\mathcal{G}$, and $\mathsf{Prof}_{\mathcal{G}}$ for the set of strategy profiles in $\mathcal{G}$.*

Notice that we only consider non-randomized (*pure*) strategies in this paper.

Let $\mathcal{G}$ be a concurrent game, P be a coalition, and σ_P be a strategy for P. A path $\pi = (s_j)_{0\leq j\leq|\pi|}$ is *compatible* with the strategy σ_P if, for all $k \leq |\pi| - 1$, there exists a move m_{Agt} such that:

- $m_A \in \mathsf{Mov}(s_k, A)$ for all $A \in \mathsf{Agt}$,
- $m_A = \sigma_A(\pi_{\leq k})$ for all $A \in P$,
- $(s_k, s_{k+1}) \in \mathsf{Tab}(s_k, m_{\mathsf{Agt}})$.

We write $\mathsf{Out}_{\mathcal{G}}(\sigma_P)$ for the set of paths (also called *outcomes*) in $\mathcal{G}$ that are compatible with strategy σ_P of coalition P. We write $\mathsf{Out}^f_{\mathcal{G}}$ (resp. $\mathsf{Out}^\infty_{\mathcal{G}}$) for the finite (resp. infinite) outcomes, and $\mathsf{Out}_{\mathcal{G}}(s, \sigma_P)$, $\mathsf{Out}_{\mathcal{G}}(s, \sigma_P)$ and $\mathsf{Out}_{\mathcal{G}}(s, \sigma_P)$ for the respective sets of outcomes of σ_P with initial state s.

Notice that, in the case of deterministic concurrent games, any strategy profile has a single infinite outcome. This might not be the case for non-deterministic concurrent games.

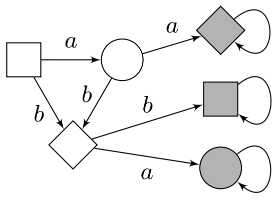

Fig. 1. A 3-player turn-based game

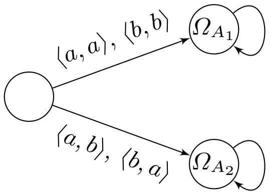

Fig. 2. A 2-player concurrent game

Example 1. Figure 1 displays an example of a three-player turn-based game. The shape of a node indicates its owner, and the goal states are those marked in grey: for instance, Player □ controls square states, and her objective is to reach ■. She cannot achieve this on her own (she has no *winning* strategy), but can achieve it with the help of Player ◇ (both should play action b).

Figure 2 is a two-player concurrent game: from the left-most state, both players choose between actions a and b, and the game goes to the top state (which is a goal state for player A_1) if they play the same action, and to the bottom state otherwise (which is a goal state for player A_2).

Given a move m_{Agt} and an action m' for some player B, we write $m_{\mathsf{Agt}}[B \mapsto m']$ for the move n_{Agt} with $n_A = m_A$ when $A \neq B$ and $n_B = m'$. This notation is extended to strategies in the natural way. In the context of non-zero-sum games, several notions of equilibria have been defined. We present here a refinement of Nash equilibria towards non-deterministic concurrent games.

Definition 3. *Let $\mathcal{G}$ be a non-deterministic concurrent game, and s be a state of $\mathcal{G}$. A* pseudo-Nash equilibrium *in $\mathcal{G}$ from s is a pair $(\sigma_{\mathsf{Agt}}, \pi)$ where $\sigma_{\mathsf{Agt}} \in \mathsf{Prof}_{\mathcal{G}}$ and $\pi \in \mathsf{Out}(s, \sigma_{\mathsf{Agt}})$ is such that for all $B \in \mathsf{Agt}$ and all $\sigma' \in \mathsf{Strat}^B$, it holds:*

$$\forall \pi' \in \mathsf{Out}(s, \sigma_{\mathsf{Agt}}[B \mapsto \sigma']).\ \pi' \preccurlyeq_B \pi.$$

Such an outcome π is called an optimal play *for the strategy profile σ_{Agt}. The* payoff *of a pseudo-Nash equilibrium $(\sigma_{\mathsf{Agt}}, \pi)$ is the function $\nu \colon \mathsf{Agt} \to \{0,1\}$ where $\nu(A) = 1$ if π visits Ω_A (the objective of Player A), and $\nu(A) = 0$ otherwise.*

In the case of deterministic games, π is uniquely determined by σ_{Agt}, and pseudo-Nash equilibria coincide with *Nash equilibria* as defined in [13]: they are strategy profiles where no player has an incentive to unilaterally deviate from her strategy.

In the case of non-deterministic games, a strategy profile for an equilibrium may give rise to several outcomes. The choice of playing the optimal play π is then made cooperatively by all players: once a strategy profile is fixed, non-determinism is resolved by all players choosing one of the possible outcomes in such a way that each player has no incentive to unilaterally changing her choice (nor her strategy). To our knowledge, this cannot be encoded by adding an extra player for solving non-determinism. Notice that solution concepts involving an extra player for solving non-determinism can be handled by our algorithm since it yields a deterministic game (leading to *real* Nash equilibria).

Example 1 (cont'd). In the (deterministic) game of Fig. 1, the strategy profile where all players play a is *not* a Nash equilibrium from □, since player ○ would better play b and reach her winning state. The profile where they all play b is a Nash equilibrium. Actually, deterministic turn-based games such as this one always admit a Nash equilibrium [7].

Now, consider the same game as depicted in Fig. 1, but in which player ◇ has only one action available, say a, which non-deterministically leads to either ■ or ●. Then none of the two outcomes □→◇→■ and □→◇→● is globally better than the other one, hence they do not correspond to a pseudo-Nash equilibrium. The reader can check that, for any strategy profile, there never exists an optimal play, so that this modified, non-deterministic turn-based game does not admit any pseudo-Nash equilibrium.

Regarding the concurrent game of Fig. 2, it is easily seen that it also does not admit a (non-randomized) Nash equilibrium.

2.2 Decision problems

In this paper we are interested in several decision problems related to the existence of pseudo-Nash equilibria. Let $\mathcal{S}$ be a class of concurrent games. In the sequel, we consider the following problems: given $\mathcal{G} = \langle \mathsf{States}, \mathsf{Edg}, \mathsf{Agt}, \mathsf{Act}, \mathsf{Mov}, \mathsf{Tab}, \preccurlyeq\rangle$ a concurrent game in class $\mathcal{S}$ with reachability objectives $\Omega_A \subseteq \mathsf{States}$ for every player A, and a state $s \in \mathsf{States}$:

Problem 1 (Existence). Does there exists a pseudo-Nash-equilibrium in $\mathcal{G}$ from s?

Problem 2 (Verification). Given a payoff $\nu\colon \mathsf{Agt} \to \{0,1\}$, does there exists a pseudo-Nash-equilibrium in $\mathcal{G}$ from s with payoff ν?

Problem 3 (Constrained-Existence). Given a constraint (given as a subset $P \subseteq \mathsf{Agt}$ and a function $\varpi\colon P \to \{0,1\}$), does there exists a pseudo-Nash-equilibrium in $\mathcal{G}$ from s with some payoff ν satisfying the constraint (*i.e.*, s.t. $\nu(A) = \varpi(A)$ for all $A \in P$)?

Notice that Problems 1 and 2 are trivially logspace-reducible to Problem 3. Together with these problems, we also consider the corresponding *function problems*: for the *verification* problem ("does the given payoff vector correspond to some equilibrium?"), the function problem asks to build a strategy profile that is an equilibrium for this payoff. For the other two problems, the function problem asks to compute a possible payoff function, and a corresponding strategy profile.

3 Qualitative Nash Equilibria

We now explain a procedure to describe pseudo-Nash equilibria in our setting. To this aim, we introduce the notion of *repellor sets*.

3.1 The repellor sets

Definition 4. *We define the set of* suspect *players for an edge $e = (s, s')$ given a move m_{Agt}, which we denote with $\mathsf{Susp}_{\mathcal{G}}(e, m_{\mathsf{Agt}})$, as the set:*

$$\{B \in \mathsf{Agt} \mid \exists m' \in \mathsf{Mov}(s, B) \text{ s.t. } e \in \mathsf{Tab}(s, m_{\mathsf{Agt}}[B \mapsto m'])\}.$$

We extend this notion to a finite path $\pi = (s_p)_{p \leq |\pi|}$ given strategies σ_{Agt} as follows:

$$\mathsf{Susp}(\pi, \sigma_{\mathsf{Agt}}) = \bigcap_{p < |\pi|} \mathsf{Susp}((s_p, s_{p+1}), (\sigma_A(\pi_{\leq p}))_{A \in \mathsf{Agt}}).$$

Intuitively, Player B is suspect for an edge e, given a move m_{Agt}, whenever she can unilaterally change her action (while the other actions are unchanged) and take edge e. Notice that if $e \in \mathsf{Tab}(s, m_{\mathsf{Agt}})$, then $\mathsf{Susp}(e, m_{\mathsf{Agt}}) = \mathsf{Agt}$. Player B is then suspect for a finite path π, given a tuple of strategies σ_{Agt}, whenever she has a strategy to enforce path π under the strategies $(\sigma_A)_{A \in \mathsf{Agt} \smallsetminus \{B\}}$ of the other players.

Lemma 5. *Given $\sigma_{\mathsf{Agt}} \in \mathsf{Prof}$ and $\pi \in \mathsf{Hist}$, the following three propositions are equivalent:*

(i) $B \in \mathsf{Susp}(\pi, \sigma_{\mathsf{Agt}})$
(ii) $\exists \sigma' \in \mathsf{Strat}^B.\ \pi \in \mathsf{Out}^f(\sigma_{\mathsf{Agt}}[B \mapsto \sigma'])$
(iii) $\pi \in \mathsf{Out}^f((\sigma_A)_{A \in \mathsf{Agt} \smallsetminus \{B\}})$

We now define the central notion of this paper, namely the *repellor sets.*

Definition 6. *Let $\mathcal{G}$ be a non-deterministic concurrent game. Given a subset $P \subseteq \mathsf{Agt}$, the* repellor set *of P, denoted by $\mathsf{Rep}_{\mathcal{G}}(P)$, is defined inductively on P as follows: as the base case, $\mathsf{Rep}_{\mathcal{G}}(\varnothing) = \mathsf{States}$; Then, assuming that $\mathsf{Rep}_{\mathcal{G}}(P')$ has been defined for all $P' \subsetneq P$, we let $\mathsf{Rep}_{\mathcal{G}}(P)$ be the largest set[2] satisfying the following two conditions:*

- $\forall A \in P.\ \mathsf{Rep}_{\mathcal{G}}(P) \cap \Omega_A = \varnothing \qquad (1)$
- $\forall s \in \mathsf{Rep}_{\mathcal{G}}(P).\ \exists m_{\mathsf{Agt}} \in \mathsf{Act}^{\mathsf{Agt}}.\ \forall s' \in \mathsf{States}.$
  $s' \in \mathsf{Rep}_{\mathcal{G}}(P \cap \mathsf{Susp}_{\mathcal{G}}((s, s'), m_{\mathsf{Agt}})) \qquad (2)$

Intuitively, from a state in $\mathsf{Rep}_{\mathcal{G}}(P)$, the players can cooperate in order to stay in this repellor set (thus never satisfying the objectives of players in P) in such a way that breaking the cooperation does not help fulfilling one's objective.

Lemma 7. *If $P \subseteq P'$, then $\mathsf{Rep}(P') \subseteq \mathsf{Rep}(P)$.*

Remark 8. Because deterministic turn-based games are determined, they enjoy the property that $\mathsf{Rep}(\{A\}) = \mathsf{States} \smallsetminus \mathsf{Win}(\{A\})$, where $\mathsf{Win}(\{A\})$ is the set of states from which player A has a winning strategy for reaching her objective against the coalition $\mathsf{Agt} \smallsetminus \{A\}$. Notice that this does not hold in concurrent games: in the game depicted on Fig. 2, the initial state is neither in the repellor set nor in the winning set of any player.

The sets of *secure moves* for staying in $\mathsf{Rep}(P)$ is defined as:

$$\mathsf{Secure}_{\mathcal{G}}(s, P) = \{(m_i)_{A_i \in \mathsf{Agt}} \in \mathsf{Act}^{\mathsf{Agt}} \mid \forall s' \in \mathsf{States}.\ s' \in \mathsf{Rep}(P \cap \mathsf{Susp}((s, s'), m_{\mathsf{Agt}}))$$

We define the transition system $\mathcal{S}_{\mathcal{G}}(P) = (\mathsf{States}, \mathsf{Edg}')$ as follows: $(s, s') \in \mathsf{Edg}'$ iff there exists some $m_{\mathsf{Agt}} \in \mathsf{Secure}(s, P)$ such that $(s, s') \in \mathsf{Tab}(s, m_{\mathsf{Agt}})$. Note in particular that any $s \in \mathsf{Rep}(P)$ has an outgoing transition in $\mathcal{S}_{\mathcal{G}}(P)$.

Example 2. In the game of Fig. 1, state ◇ is in the repellor set of {□, ◇} and of {○, ◇} but not in that of {□, ○}. Intuitively, from that state, Player ◇ can prevent one of the other two players to reach her objective, but not both of them at the same time. It can be checked that Rep({□, ◇}) = {□; ○; ◇; ●}.

Looking now at the same game but with non-determinism in state ◇, the repellor sets are different; in particular, state ◇ is no longer in Rep({□}) nor in Rep({○}).

[2] This is uniquely defined since if two sets satisfy both conditions, then so does their union.

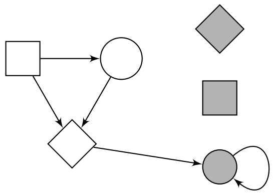

Fig. 3. $\mathcal{S}(\{\square,\diamond\})$ for the det. game

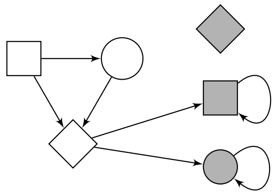

Fig. 4. $\mathcal{S}(\{\diamond\})$ for the non-det. game

3.2 Using the repellor to characterize (pseudo-)Nash equilibria

We now draw the link between the repellor sets and (pseudo-)Nash equilibria.

Lemma 9. *Let $P \subseteq \mathsf{Agt}$, and $s \in \mathsf{States}$. Then $s \in \mathsf{Rep}(P)$ if and only if there exists an infinite path π in $\mathcal{S}(P)$ starting from s.*

Repellor sets characterize those states from which one can find equilibria that avoid the reachability objectives of players in P:

Proposition 10. *Let $P \subseteq \mathsf{Agt}$, and $\pi \in \mathsf{Play}(s)$ be an infinite play with initial state s. Then π is a path in $\mathcal{S}(P)$ if and only if there exists $\sigma_{\mathsf{Agt}} \in \mathsf{Prof}$ such that $\pi \in \mathsf{Out}(s, \sigma_{\mathsf{Agt}})$ and for all $B \in P$ and all $\sigma' \in \mathsf{Strat}^B$ it holds:*

$$\forall \pi \in \mathsf{Out}(s, \sigma_{\mathsf{Agt}}[B \mapsto \sigma']).\ \pi \textit{ does not visit } \Omega_B.$$

Theorem 11. *Let $\mathcal{G} = \langle \mathsf{States}, \mathsf{Edg}, \mathsf{Agt}, \mathsf{Act}, \mathsf{Mov}, \mathsf{Tab}, \preccurlyeq\rangle$ be a concurrent game, with reachability objectives $\Omega_A \subseteq \mathsf{States}$ for each player $A \in \mathsf{Agt}$, and $s \in \mathsf{States}$. There is a pseudo-Nash equilibrium from s with payoff ν iff, letting P be the set $\{A \in \mathsf{Agt} \mid \nu(A) = 0\}$, there is an infinite path π in $\mathcal{S}(P)$ which starts in s and which visits Ω_A for every A not in P. Furthermore, π is the optimal play of some pseudo-Nash equilibrium.*

This gives a generic procedure to decide the existence of pseudo-Nash equilibria in non-deterministic concurrent games. It is not effective yet (remember that we allow uncountably-infinite games), but will yield algorithms when instantiated on finite games and timed games in the forthcoming sections.

Proof (of Theorem 11). ($\Rightarrow$) Let $(\sigma_{\mathsf{Agt}}, \pi)$ be a pseudo-Nash equilibrium: σ_{Agt} is a strategy profile, and $\pi \in \mathsf{Out}(s, \sigma_{\mathsf{Agt}})$ is such that for any player B and any strategy σ' for B, it holds

$$\forall \pi' \in \mathsf{Out}(s, \sigma_{\mathsf{Agt}}[B \mapsto \sigma']).\ \pi' \text{ visits } \Omega_B \Rightarrow \pi \text{ visits } \Omega_B.$$

Moreover, π visits Ω_B iff $B \notin P$. According to Proposition 10, π must be a path in the transition system $\mathcal{S}(P)$.

($\Leftarrow$) Let π be an infinite path in $\mathcal{S}(P)$ such that for every $B \notin P$, π visits some state in Ω_B. According to Proposition 10, there is a strategy profile such that π is one of its outcomes and if any player $A \in P$ deviates, no outcome visits Ω_A. Together with π, this forms a pseudo-Nash equilibrium. □

Theorem 11 gives a (necessary and sufficient) condition for the existence of a pseudo-Nash equilibrium in a game. In case an equilibrium exists, repellor sets (and the corresponding transition systems) also contain all the necessary information for effectively computing a pseudo-Nash equilibrium:

Proposition 12. *If π is an infinite path in $\mathcal{S}(P)$ from s visiting Ω_B for every $B \notin P$, then there is a pseudo-Nash equilibrium $(\sigma_{\mathsf{Agt}}, \pi)$ where strategies σ_{Agt} consist in playing secure moves in the transition system $\mathcal{S}(P \cap P')$, for some P'.*

Proof (Sketch). The strategy profile should contain π as one of its outcomes, which can be done by selecting the relevant moves from $\mathcal{S}(P)$. Now, if the play ever gets outside of π but still in $\mathsf{Rep}(P)$ (which may occur because of non-determinism, or because some player, who would have won, has deviated from her strategy), then the strategy profile should select secure moves to stay in this set. Finally, if the history exits $\mathsf{Rep}(P)$, this indicates that (at least) one player in P is trying to deviate from her selected strategy. The strategy profile must ensure that she cannot win: this is achieved by detecting the set P' of players who may be responsible for the deviation, and play secure moves in $\mathcal{S}(P \cap P')$. □

Example 3. For the game of Fig. 1, consider for instance the transition system $\mathcal{S}(\{\square, \diamond\})$, which is depicted on Fig. 3. There are two infinite paths from state □; they correspond to the outcomes of the two Nash equilibria in the game of Fig. 1, both of which have payoff $(\square \mapsto 0, \bigcirc \mapsto 1, \Diamond \mapsto 0)$.

In the same game with non-determinism in state ◇, the transition system $\mathcal{S}(\{\square, \diamond\})$ can be checked to contain no edges, so that there is no pseudo-Nash equilibria with payoff $(\square \mapsto 0, \bigcirc \mapsto 1, \Diamond \mapsto 0)$. Now, if we look at $\mathcal{S}(\{\diamond\})$, which is depicted at Fig. 4, there are four possible infinite paths in this transition system, but none of them visits both ■ and ●. It does not give us a pseudo-Nash equilibrium and in fact there is none in this game.

4 Application to finite games

In this section, we apply the previous generic procedure to finite concurrent games. We consider four classes of finite concurrent games: $\mathfrak{C}^{\mathrm{nd}}$ is the whole class of finite concurrent non-deterministic games, $\mathfrak{C}^{\mathrm{d}}$ is the restriction to deterministic games, $\mathfrak{TB}^{\mathrm{nd}}$ is the restriction to turn-based games, and $\mathfrak{TB}^{\mathrm{d}}$ is the intersection of $\mathfrak{C}^{\mathrm{d}}$ and $\mathfrak{TB}^{\mathrm{nd}}$.

We also consider subclasses where the number of players is bounded *a priori*, and thus is not taken into account in the complexity. Our results can be summarized as follows (in grey are previously known results [7, Corollary 1][3]):

[3] The results in [17] concern parity objectives, and do not encompass reachability objectives.

	$\mathfrak{C}^{\mathrm{nd}}$, $\mathfrak{C}^{\mathrm{d}}$, $\mathfrak{TB}^{\mathrm{nd}}$		$\mathfrak{TB}^{\mathrm{d}}$	
	bounded	general	bounded	general
Existence	P-c.	NP-c.	True	True
Verification	P-c.	NP-c.	P-c.	NP-c.
Constr. Ex.	P-c.	NP-c.	P-c.	NP-c.

These results all follow from the following Proposition:

Proposition 13. *1. The following problems are* P*-hard with bounded number of players and* NP*-hard in the general case:*

(a) checking that a payoff ν corresponds to a Nash equilibrium in $\mathfrak{TB}^{d}$;

(b) deciding the existence of a pseudo-Nash equilibrium in $\mathfrak{TB}^{nd}$;

(c) deciding the existence of a Nash equilibrium in $\mathfrak{C}^{d}$.

2. Solving the constrained-existence problem in $\mathfrak{C}^{nd}$ is in P *for a bounded number of players, and in* NP *in the general case.*

Proof (Sketch of proof). P- and NP-hardness results are obtained by straightforward encodings of the CIRCUIT-VALUE and 3SAT problems, respectively.

The NP algorithm for the constrained existence problem is obtained by first guessing the payoff function, and then checking that Theorem 11 holds. This is achieved by guessing a sequence of states in $\mathcal{S}(P)$, and checking that it is indeed a path in $\mathcal{S}(P)$ and that it visits the correct sets in Ω_{Agt}. A naive implementation of this procedure runs in exponential time (because computing $\mathcal{S}(P)$ may require the computation of intermediate sets $\mathsf{Rep}(P \cap P')$ for many subsets P' of Agt, which may result in up to $2^{|P|}$ computation steps), but using non-determinism, we can select polynomially many intermediate repellor sets that must be computed. The procedure thus runs in non-deterministic polynomial time.

In the case where the number of agents is bounded, the naive approach above is already polynomial, and the number of payoff functions is also polynomial. We can then enumerate all payoff functions, build the transition system $\mathcal{S}(P)$ for each of them, and check the existence of a "witness" path in this transition system. □

Remark 14. In the case of turn-based games, the set of suspects is always either empty, or a singleton, or the whole set of players. As a consequence, the naive implementation of the procedure above will not result in computing $2^{|P|}$ repellor sets, but only $|P|$. The global algorithm still runs in NP, because finding a path in $\mathcal{S}(P)$ with several reachability constraints is NP-complete.

5 Application to timed games

5.1 Definitions of timed games

A *valuation* over a finite set of clocks X is an application $v\colon X \to \mathbb{R}_+$. If v is a valuation and $t \in \mathbb{R}_+$, then $v + t$ is the valuation that assigns to each $x \in X$ the value $v(x)+t$. If v is a valuation and $Y \subseteq X$, then $[Y \leftarrow 0]v$ is the valuation that assigns 0 to each $y \in Y$ and $v(x)$ to each $x \in X \setminus Y$. A *clock constraint* over X is a formula built on the grammar: $\mathfrak{C}(X) \ni g \ ::= \ x \sim c \ \mid \ g \wedge g$, where x ranges over X, $\sim \in \{<, \leq, =, \geq, >\}$, and c is an integer. The semantics of clock constraints over valuations is natural, and we omit it.

We now define the notion of timed games that we will use in this paper. Our definition follows that of [8].

Definition 15. *A* timed game *is a 7-tuple* $\mathcal{G} = \langle \mathit{Loc}, X, \mathit{Inv}, \mathit{Trans}, \mathit{Agt}, \mathit{Owner}, \preccurlyeq \rangle$ *where:*

- Loc *is a finite set of locations;*
- X *is a finite set of* clocks*;*
- $\mathit{Inv}\colon \mathit{Loc} \to \mathfrak{C}(X)$ *assigns an* invariant *to each location;*
- $\mathit{Trans} \subseteq \mathit{Loc} \times \mathfrak{C}(clocks) \times 2^X \times \mathit{Loc}$ *is the set of transitions;*
- Agt *is a finite set of agents (or players);*
- $\mathit{Owner}\colon \mathit{Trans} \to \mathit{Agt}$ *assigns an agent to each transition;*
- $\preccurlyeq\colon \mathit{Agt} \to 2^{(\mathit{States} \times \mathbb{R}_+)^\omega \times (\mathit{States} \times \mathbb{R}_+)^\omega}$ *defines, for each player, a quasi-ordering on the runs of* $\mathcal{G}$*, called* preference relation*.*

As in the previous sections, we restrict here to the case where $\preccurlyeq$ is given in terms of reachability objectives $(\Omega_A)_{A \in \mathsf{Agt}}$, with $\Omega_A \subseteq \mathsf{Loc}$ for each $A \in \mathsf{Agt}$.

A timed game is played as follows: a state of the game is a pair (ℓ, v) where ℓ is a location and v is a clock valuation, provided that $v \models \mathsf{Inv}(\ell)$. From each state (starting from an initial state $s_0 = (\ell, \mathbf{0})$, where $\mathbf{0}$ maps each clock to zero and is assumed to satisfy $\mathsf{Inv}(\ell)$), each player A chooses a nonnegative real number d and a transition δ, with the intended meaning that she wants to delay for d time units and then fire transition δ. There are several (natural) restrictions on these choices:

- spending d time units in ℓ must be allowed[4] *i.e.*, $v + d \models \mathsf{Inv}(\ell)$;
- $\delta = (\ell, g, z, \ell')$ belongs to player A, *i.e.*, $\mathsf{Owner}(\delta) = A$;
- the transition is firable after d time units (*i.e.*, $v + d \models g$), and the invariant is satisfied when entering ℓ' (*i.e.*, $[z \leftarrow 0](v + d) \models \mathsf{Inv}(\ell')$).

If (and only if) there is no such possible choice for some player A (*e.g.* if no transition from ℓ belongs to A), then she chooses a special move, denoted by $\perp$.

Given the set of choices m_{Agt} of all the players, with $m_A \in (\mathbb{R}_+ \times \mathsf{Trans}) \cup \{\perp\}$, a player B such that $d_B = \min\{d_A \mid A \in \mathsf{Agt} \text{ and } m_A = (d_A, \delta_A)\}$ is selected (non-deterministically), and the corresponding transition δ_B is applied, leading to a new state $(\ell', [z \leftarrow 0](v + d))$.

This semantics can naturally be expressed in terms of an infinite-state non-deterministic concurrent game. Timed games inherit the notions of history, play, path, strategy, profile, outcome and (pseudo-)Nash equilibrium *via* this correspondence.

[4] Formally, this should be written $v + d' \models \mathsf{Inv}(\ell)$ for all $0 \leq d' \leq d$, but this is equivalent to having only $v \models \mathsf{Inv}(\ell)$ and $v + d \models \mathsf{Inv}(\ell)$ since invariants are convex.

In the sequel, we consider only *non-blocking* timed games, *i.e.*, timed games in which, for any reachable state (ℓ, v), at least one player has an allowed action:

$$\prod_{A \in \mathsf{Agt}} \mathsf{Mov}((\ell, v), A) \neq \{(\bot)_{A \in \mathsf{Agt}}\}.$$

5.2 Computing pseudo-Nash equilibria in timed games

Let $\mathcal{G} = \langle \mathsf{Loc}, X, \mathsf{Inv}, \mathsf{Trans}, \mathsf{Agt}, \mathsf{Owner}, \preccurlyeq \rangle$ be a timed game, where $\preccurlyeq$ is given in terms of reachability objectives $(\Omega_A)_{A \in \mathsf{Agt}}$. In this section, we explain how pseudo-Nash equilibria can be computed in such reachability timed games, using Theorem 11. This relies on the classical notion of *regions* [1], which we assume the reader is familiar with.

We define the region game $\mathcal{R} = \langle \mathsf{States}_{\mathcal{R}}, \mathsf{Edg}_{\mathcal{R}}, \mathsf{Agt}, \mathsf{Act}_{\mathcal{R}}, \mathsf{Mov}_{\mathcal{R}}, \mathsf{Tab}_{\mathcal{R}}, \preccurlyeq_{\mathcal{R}} \rangle$ as follows:

- $\mathsf{States}_{\mathcal{R}} = \{(\ell, r) \in \mathsf{Loc} \times \mathfrak{R} \mid r \models \mathsf{Inv}(\ell)\}$, where $\mathfrak{R}$ is the set of clock regions;
- $\mathsf{Edg}_{\mathcal{R}}$ is the set of transitions of the region automaton underlying $\mathcal{G}$;
- $\mathsf{Act}_{\mathcal{R}} = \{(r, p, \delta) \mid r \in \mathfrak{R},\ p \in \{1; 2; 3\} \text{ and } \delta \in \mathsf{Trans}\} \cup \{\bot\}$;
- $\mathsf{Mov}_{\mathcal{R}} \colon \mathsf{States}_{\mathcal{R}} \times \mathsf{Agt} \to 2^{\mathsf{Act}_{\mathcal{R}}} \setminus \{\varnothing\}$ is such that

  $$\begin{aligned} \mathsf{Mov}_{\mathcal{R}}((\ell, r), A) = \{(r', p, \delta) \mid {} & r' \in \mathsf{Succ}(r),\ r' \models \mathsf{Inv}(\ell), \\ & p \in \{1; 2; 3\} \text{ if } r' \text{ is time-elapsing, else } p = 1, \\ & \delta = (\ell, g, z, \ell') \in \mathsf{Trans} \text{ is such that } r' \models g \\ & \qquad \text{and } [z \leftarrow 0]r' \models \mathsf{Inv}(\ell')\text{and } \mathsf{Owner}(\delta) = A\} \end{aligned}$$

  if it is non-empty, and $\mathsf{Mov}_{\mathcal{R}}((\ell, r), A) = \{\bot\}$ otherwise. Roughly, the index p allows the players to say if they want to play first, second or later if their region is selected.
- $\mathsf{Tab}_{\mathcal{R}} \colon \mathsf{States}_{\mathcal{R}} \times {\mathsf{Act}_{\mathcal{R}}}^{\mathsf{Agt}} \to 2^{\mathsf{Edg}_{\mathcal{R}}} \setminus \{\varnothing\}$ is such that for every $(\ell, r) \in \mathsf{States}_{\mathcal{R}}$ and every $m_{\mathsf{Agt}} \in \prod_{A \in \mathsf{Agt}} \mathsf{Mov}_{\mathcal{R}}((\ell, r), A)$, if we write r' for[5] $\min\{r_A \mid m_A = (r_A, p_A, \delta_A)\}$ and p' for $\min\{p_A \mid m_A = (r', p_A, \delta_A)\}$,

  $$\begin{aligned} \mathsf{Tab}_{\mathcal{R}}((\ell, r), m_{\mathsf{Agt}}) = \{((\ell, r), (\ell_B, [z_B \leftarrow 0] r_B)) \mid \\ m_B = (r_B, p_B, \delta_B) \text{ with } r_B = r',\ p_B = p' \text{ and } \delta_B = (\ell, g_B, z_B, \ell_B)\}. \end{aligned}$$

- The preference relation $\preccurlyeq_{\mathcal{R}}$ is defined in terms of reachability objectives for each player, where the set of objectives $(\Omega'_i)_{A_i \in \mathsf{Agt}}$ $(\Omega'_A)_{A \in \mathsf{Agt}}$ is defined, for each $A \in \mathsf{Agt}$, as $\Omega'_A = \{(\ell, r) \mid \ell \in \Omega_A, r \in \mathfrak{R}\}$.

Proposition 16. *Let $\mathcal{G}$ be a timed game, and $\mathcal{R}$ its associated region game. Then there is a pseudo-Nash equilibrium in $\mathcal{G}$ from $(s, \mathbf{0})$ iff there is a pseudo-Nash equilibria in $\mathcal{R}$ from $(s, [\mathbf{0}])$, where $[\mathbf{0}]$ is the region associated to $\mathbf{0}$. Furthermore, this equivalence is constructive.*

[5] This is well-defined for two reasons: first, not all m_i's may be $\bot$, since we consider non-blocking games; second, the set of regions appearing in a move from (ℓ, r) only contains successors of r, and is then totally ordered.

Proof (Sketch of proof). The proof is in three steps: we first define a kind of generic *simulation* relation between games, which gives information on their respective repellor sets and transition systems:

Lemma 17. *Consider two games $\mathcal{G}$ and $\mathcal{G}'$ involving the same set of agents, with preference relations defined in terms of reachability conditions $(\Omega_A)_{A \in \mathsf{Agt}}$ and $(\Omega'_A)_{A \in \mathsf{Agt}}$, respectively. Assume that there exists a binary relation $\lhd$ between states of $\mathcal{G}$ and states of $\mathcal{G}'$ such that, if $s \lhd s'$, then:*

1. *if $s' \in \Omega'_A$ then $s \in \Omega_A$ for any $A \in \mathsf{Agt}$;*
2. *for all move m_{Agt} in $\mathcal{G}$, there exists a move m'_{Agt} in $\mathcal{G}'$ such that:*
   - *for any t' in $\mathcal{G}'$, there is $t \lhd t'$ in $\mathcal{G}$ s.t. $\mathsf{Susp}((s',t'), m'_{\mathsf{Agt}}) \subseteq \mathsf{Susp}((s,t), m_{\mathsf{Agt}})$;*
   - *for any (s,t) in $\mathsf{Tab}(s, m_{\mathsf{Agt}})$, there is a (s',t') in $\mathsf{Tab}(s', m'_{\mathsf{Agt}})$ s.t. $t \lhd t'$.*

Then for any $P \subseteq \mathsf{Agt}$ and for any s and s' such that $s \lhd s'$, it holds:

1. *if s is in $\mathsf{Rep}_{\mathcal{G}}(P)$, then s' is in $\mathsf{Rep}_{\mathcal{G}'}(P)$;*
2. *for any $(s,t) \in \mathsf{Edg}_{\mathsf{Rep}}$, there exists (s',t') in $\mathsf{Edg}'_{\mathsf{Rep}}$ s.t. $t \lhd t'$, where $\mathsf{Edg}_{\mathsf{Rep}}$ and $\mathsf{Edg}'_{\mathsf{Rep}}$ are the set of edges in the transition systems $\mathcal{S}_{\mathcal{G}}(P)$ and $\mathcal{S}_{\mathcal{G}'}(P)$, respectively.*

It remains to show that a timed game and its associated region game simulate one another in the sense of Lemma 17, which entail that they have the same sets of repellors. This is achieved by defining two functions λ, mapping moves in $\mathcal{G}$ to *equivalent* moves in $\mathcal{R}$, and μ, mapping moves in $\mathcal{R}$ to *equivalent* moves in $\mathcal{G}$, in such a way that $v \lhd r$ iff r is the region containing v. Theorem 11 concludes the proof. □

Because the region game $\mathcal{R}$ has size exponential in the size of $\mathcal{G}$, we get:

Theorem 18. *The constrained existence problem (and thus the existence- and verification problems) in timed game can be solved in $\mathsf{EXPTIME}$.*

Remark 19. Given a pseudo-Nash equilibrium $(\alpha_{\mathsf{Agt}}, \pi)$ in the region game, we can obtain one in the timed game for the same payoff vector. Assume that $(\alpha_{\mathsf{Agt}}, \pi)$ is a pseudo-Nash equilibrium in $\mathcal{R}$. Given a history h in $\mathcal{G}$ and its projection $\mathsf{proj}(h)$ in $\mathcal{R}$, if $(\alpha_A(\mathsf{proj}(h)))_{A \in \mathsf{Agt}} = (r_A, p_A, \delta_A)_{A \in \mathsf{Agt}}$ is a secure move in $\mathcal{R}$, then so is $(\mu_A)_{A \in \mathsf{Agt}} = \mu(\mathsf{last}(h), (\alpha_A(\mathsf{proj}(h)))_{A \in \mathsf{Agt}})$, where μ is the function used in the proof of Proposition 16 to simulate moves from $\mathcal{R}$ in $\mathcal{G}$. Moreover, there exists a play $\pi' \in \mathsf{Out}((\ell, v), (\mu_A)_{A \in \mathsf{Agt}})$ such that $\mathsf{proj}(\pi') = \pi$, therefore the payoff function for these two plays is the same. Hence $((\mu_A)_{A \in \mathsf{Agt}}), \pi')$ is a pseudo-Nash equilibrium in the timed game.

Our algorithm is optimal, as we prove $\mathsf{EXPTIME}$-hardness of our problems:

Proposition 20. *The constrained-existence and verification problems for deterministic turn-based timed games with at least two clocks and two players is $\mathsf{EXPTIME}$-hard. The existence problem is $\mathsf{EXPTIME}$-hard for concurrent timed games (with at least two clocks and two players).*

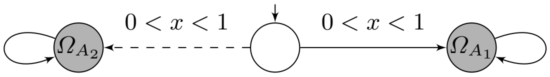

Fig. 5. A timed game with no equilibria (the solid transition belongs to the first player, the dashed one to the second one).

This is proved by encoding countdown games [11]. The second part of the Proposition requires the use of a timed game with no equilibria; an example of such a game is depicted on Fig. 5.

Remark 21. Since deterministic turn-based timed games yield deterministic turn-based region games, they always admit a Nash equilibrium.

6 Conclusion

In this paper we have described a procedure to compute qualitative pseudo-Nash equilibria in multi-player concurrent (possibly non-deterministic) games with reachability objectives. The development of this procedure has required technical tools as the repellor sets, which can be seen as an alternative to the classical attractor sets for computing equilibria in games. We have applied this procedure to finite concurrent games and to timed games, yielding concrete algorithms to compute equilibria in those games. We have furthermore proved that those algorithms have optimal complexities.

Multiple extensions of this work are rather natural:

- First we would like to apply the generic procedure to other classes of systems, for instance to pushdown games [20]. Note that we are not aware of any result on the computation of equilibria in pushdown games.
- Then our procedure only applies to reachability objectives for every player. It would be interesting to adapt it to other ω-regular winning objectives. This is *a priori* non-trivial as this will require developing new tools (the repellor sets are dedicated to reachability objectives).
- We have applied our procedure to concurrent games as defined *e.g.* in [2], where the transition table of the game is given in extensive form (for each tuple of possible actions, there is an entry in a table). In [12], a more compact way of representing concurrent game is proposed, which assumes a symbolic representation of the transition table. It would be interesting to study how this does impact on the complexity of the computation of Nash equilibria. In particular the argument for having an NP algorithm (Proposition 13) does not hold anymore.
- Finally other notions of equilibria (subgame-perfect equilibria, secure equilibria, *etc*) could be investigated, and extensions of concurrent games with probabilities could also be investigated.

References

1. R. Alur and D. L. Dill. A theory of timed automata. *TCS*, 126(2):183–235, 1994.
2. R. Alur, T. A. Henzinger, and O. Kupferman. Alternating-time temporal logic. *J. ACM*, 49:672–713, 2002.
3. E. Asarin, O. Maler, A. Pnueli, and J. Sifakis. Controller synthesis for timed automata. In *Proc. IFAC Symp. System Structure and Control*, pages 469–474. Elsevier, 1998.
4. G. Behrmann, A. Cougnard, A. David, E. Fleury, K. G. Larsen, and D. Lime. UPPAAL-Tiga: Time for playing games! In *CAV'07*, volume 4590 of *LNCS*, pages 121–125. Springer, 2007.
5. P. Bouyer, R. Brenguier, and N. Markey. Nash equilibria for reachability objectives in multi-player timed games. research report LSV-10-12, Lab. Spécification & Vérification, ENS Cachan, France, June 2010.
6. K. Chatterjee, T. A. Henzinger, and M. Jurdziński. Games with secure equilibria. In *LICS'06*, pages 160–169. IEEE Comp. Soc. Press, 2006.
7. K. Chatterjee, R. Majumdar, and M. Jurdziński. On Nash equilibria in stochastic games. In *CSL'04*, volume 3210 of *LNCS*, pages 26–40. Springer, 2004.
8. L. de Alfaro, M. Faella, T. A. Henzinger, R. Majumdar, and M. Stoelinga. The element of surprise in timed games. In *CONCUR'03*, volume 2761 of *LNCS*, pages 142–156. Springer, 2003.
9. M. Félegyházi, J.-P. Hubaux, and L. Buttyán. Nash equilibria of packet forwarding strategies in wireless ad hoc networks. *IEEE Trans. Mobile Computing*, 5(5):463–476, 2006.
10. T. A. Henzinger. Games in system design and verification. In *TARK'05*, pages 1–4. Nat. Univ. Singapore, 2005.
11. M. Jurdziński, F. Laroussinie, and J. Sproston. Model checking probabilistic timed automata with one or two clocks. In *TACAS'07*, volume 4424 of *LNCS*, pages 170–184. Springer, 2007.
12. F. Laroussinie, N. Markey, and G. Oreiby. On the expressiveness and complexity of ATL. *LMCS*, 4(2:7), 2008.
13. J. F. Nash. Equilibrium points in n-person games. *Proc. National Academy of Sciences of the USA*, 36(1):48–49, 1950.
14. S. Paul and S. Simon. Nash equilibrium in generalised Muller games. In *FSTTCS'09*, volume 4 of *LIPIcs*, pages 335–346. LZI, 2009.
15. W. Thomas. Infinite games and verification (extended abstract of a tutoral). In *CAV'02*, volume 2404 of *LNCS*, pages 58–64. Springer, 2002.
16. M. Ummels. Rational behaviour and strategy construction in infinite multiplayer games. In *FSTTCS'06*, volume 4337 of *LNCS*, pages 212–223. Springer, 2006.
17. M. Ummels. The complexity of Nash equilibria in infinite multiplayer games. In *FoSSaCS'08*, volume 4962 of *LNCS*, pages 20–34. Springer, 2008.
18. M. Ummels and D. Wojtczak. The complexity of Nash equilibria in simple stochastic multiplayer games. In *ICALP'09*, volume 5556 of *LNCS*, pages 297–308. Springer, 2009.
19. M. Ummels and D. Wojtczak. Decision problems for Nash equilibria in stochastic games. In *CSL'09*, volume 5771 of *LNCS*, pages 515–530. Springer, 2009.
20. I. Walukiewicz. Pushdown processes: Games and model checking. In *CAV'96*, volume 1102 of *LNCS*, pages 234–263. Springer, 1996.