As Soon as Possible but Rationally

Véronique Bruyère
Université de Mons (UMONS), Belgium

Christophe Grandmont
Université de Mons (UMONS), Belgium
Université Libre de Bruxelles (ULB), Belgium

Jean-François Raskin
Université Libre de Bruxelles (ULB), Belgium

Abstract

This paper addresses complexity problems in rational verification and synthesis for multi-player games played on weighted graphs, where the objective of each player is to minimize the cost of reaching a specific set of target vertices. In these games, one player, referred to as the system, declares his strategy upfront. The other players, composing the environment, then rationally make their moves according to their objectives. The rational behavior of these responding players is captured through two models: they opt for strategies that either represent a Nash equilibrium or lead to a play with a Pareto-optimal cost tuple.

2012 ACM Subject Classification Software and its engineering → Formal methods; Theory of computation → Solution concepts in game theory; Theory of computation → Logic and verification

Keywords and phrases Games played on graphs, rational verification, rational synthesis, Nash equilibrium, Pareto-optimality, quantitative reachability objectives

Digital Object Identifier 10.4230/LIPIcs.CONCUR.2024.14

Related Version *Full Version*: `https://arxiv.org/abs/2403.00399` [17]

Funding This work has been supported by the Fonds de la Recherche Scientifique – FNRS under Grant n° T.0023.22 (PDR Rational).
Jean-François Raskin: Supported by Fondation ULB (`https://www.fondationulb.be/en/`).

1 Introduction

Nowadays, formal methods play a crucial role in ensuring the reliability of critical computer systems. Still, the application of formal methods on a large scale remains elusive in certain areas, notably in multi-agent systems. Those systems pose a significant challenge for formal verification and automatic synthesis because of their heterogeneous nature, encompassing everything from conventional reactive code segments to fully autonomous robots and even human operators. Crafting formal models that accurately represent the varied components within these systems is often a too complex task.

Although constructing detailed operational models for humans or sophisticated autonomous robots might be problematic, it is often more feasible to identify the *overarching goals* that those agents pursue. Incorporating these goals is crucial in the design and validation process of systems that interact with such entities. Typically, a system is not expected to function flawlessly under all conditions but rather in scenarios where the agents it interacts with act in alignment with their objectives, i.e., they *behave rationally*. *Rational synthesis* focuses on creating a system that meets its specifications against any behavior of environmental agents that is guided by their goals (and not against any of their behaviors). *Rational verification* studies the problem of ensuring that a system enforces certain correctness properties, not in all conceivable scenarios, but specifically in scenarios where environmental agents behave in accordance with their objectives.

35th International Conference on Concurrency Theory (CONCUR 2024).
Editors: Rupak Majumdar and Alexandra Silva; Article No. 14; pp. 14:1–14:20
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Table 1 Summary of complexity results.

	Non-coop. verif.	**Universal non-coop. verif.**	**Coop. synthesis**	**Non-coop. synthesis**
PO, weights	Π_2^{P}-complete	PSPACE-complete	PSPACE-complete	NEXPTIME-complete [11]
PO, qualitative	Π_2^{P}-complete	PSPACE-complete	PSPACE-complete	NEXPTIME-complete [18]
NE, weights	coNP-complete	coNP-complete	NP-complete	Unknown, EXPTIME-hard[1)]
NE, qualitative	coNP-complete [27]	coNP-complete [27]	NP-complete [21]	PSPACE-complete [21]

1) For the important special case of one-player environments, we provide an algorithm that runs in EXPTIME and we can prove PSPACE-hardness. The EXPTIME-hardness of the general case already holds for two-player environments.

Rationality can be modeled in various ways. In this paper, we focus on two general approaches. The first approach comes from game theory where rationality is modeled by the concept of equilibrium, such as *Nash equilibria* (NE) [35] or *subgame perfect equilibria* (SPE), a refinement of NEs [36]. The second approach treats the environment as a single agent but with multiple, sometimes conflicting, goals, aiming for behaviors that achieve a *Pareto-optimal* balance among these objectives. The concept of Pareto-optimality (PO) and its application in multi-objective analysis have been explored primarily in the field of optimization [37], but also in formal methods [2, 4]. These two notions of rationality are different in nature: in the first setting, each component of the environment playing an equilibrium is considered to be an independent selfish individual, excluding cooperation scenarios, while in the second setting, several components of the environment can collaborate and agree on trade-offs. The challenge lies in adapting these concepts to *reactive systems* characterized by ongoing, non-terminating interactions with their environment. This necessitates the transition from two-player zero-sum games on graphs, the classical approach used to model a fully adversarial environment (see e.g. [38]), to the more complex setting of *multi-player non zero-sum games on graphs* used to model environments composed of various rational agents.

Rational synthesis and rational verification have attracted large attention recently, see e.g. [7, 19, 21, 26, 28, 29, 33, 34]. But the results obtained so far, with a few exceptions that we detail below, are limited to the *qualitative* setting formalized as Boolean outcomes associated with ω-regular objectives. Those objectives are either specified using linear temporal specifications or automata over infinite words (like parity automata). The complexity of those problems is now well understood (with only a few complexity gaps remaining, see e.g. [21, 34]). The methods to solve those problems and get completeness results for worst-case complexity are either based on automata theory (using mainly automata over infinite trees) or by reduction to zero-sum games. *Quantitative* objectives are less studied and revealed to be much more challenging. For instance, it is only very recently that the rational verification problem was studied, for SPEs in non zero-sum games with mean-payoff, energy, and discounted-sum objectives in [7], for an LTL specification in multi-agent systems that behave according to an NE with mean-payoff objectives in [29] or with quantitative probabilistic LTL objectives in [30]. In [1], the rational synthesis problem was studied for the quantitative extension LTL[$\mathcal{F}$] of LTL where the Boolean operators are replaced with arbitrary functions mapping binary tuples into the interval $[0, 1]$.

In this paper, we consider *quantitative reachability* objectives. Our choice for studying these objectives was guided by their fundamental nature and also by their relative simplicity. Nevertheless, as we will see, they are challenging for both rational synthesis and rational verification. Indeed, to obtain worst-case optimal algorithms and establish completeness results, we had to resort to the use of *innovative* theoretical tools, more advanced than those necessary for the qualitative framework. In our endeavor, we have established the exact complexity of most studied decision problems in rational synthesis and rational verification.

Technical Contributions. In this work, we explore both verification and synthesis problems through the lens of rationality, defined by Pareto-optimality and Nash equilibria, for quantitative reachability objectives. For the synthesis problem, we also consider the *cooperative* variant where the environment cooperates with the system: we want to decide whether the system has a strategy and the environment a rational response to this strategy such that the objective of the system is enforced. Our results are presented in Table 1, noting that all results lacking explicit references are, to our knowledge, novel contributions. For completeness, the table includes (new and known) results for the qualitative scenario.

The results for *PO rationality* are as follows. (1) For the verification problems, we assume that the behavior of the system is formalized by a *nondeterministic Mealy machine*, used to represent a (usually infinite) set of its possible implementations. For each of those implementations, we verify that the quantitative reachability objective of the system is met against any rational behavior of the environment. We establish that this problem is PSPACE-complete. To obtain the upper bound, we rely on a *genuine combination of techniques* based on Parikh automata and a recursive PSPACE algorithm (for positive Boolean combinations of bounded safety objectives, a problem of independent interest). Parikh automata are used to guess a compact representation of certificates which are paths of possibly exponential length in the size of the problem input. When the Mealy machine is deterministic, we show that the complexity goes down to Π_2^{P}-completeness, as the previous PSPACE algorithm is replaced by a coNP oracle. (2) For the synthesis problems, we only consider the cooperative version which we prove to be PSPACE-complete, as the non-cooperative version was established to be NEXPTIME-complete in [11].

The results for *NE rationality* are as follows. (1) We establish that, surprisingly, the verification problems are coNP-complete both for the general case of a nondeterministic Mealy machine and for the special case where it is deterministic. The upper bounds for those problems are again based on Parikh automata certificates but here there is no need to use a coNP oracle. (2) For the synthesis problems, the landscape is more challenging. For the cooperative case, we were able to establish that the problem is NP-complete. For the non-cooperative case, we have partially solved the problem and established the following results. When the environment is composed of a single rational player, the problem is in EXPTIME and PSPACE-hard. For an environment with at least two players, we show that the problem is EXPTIME-hard but we leave its decidability open. The lower bounds are obtained using an elegant reduction from countdown games [31]. We give indications in the paper why the problem is difficult to solve and why classical automata-theoretic methods *may not be sufficient* (if the problem is decidable).

In this paper, we focus on nonnegative weights as we show that considering arbitrary weights leads to undecidability of the synthesis problems. We also focus on NEs instead of SPEs, even if the latter are a better concept to model rational behavior in games played on graphs. Indeed, it is well-known that SPEs pose greater challenges than NEs. So, starting with NEs offers a better initial step for the algorithmic treatment of rational verification and synthesis in quantitative scenarios, an area that remains largely unexplored.

Related Work. The survey [15] presents several results about different game models and different kinds of objectives related to reachability. Quantitative objectives in *two-player zero-sum games* were largely studied, see e.g. [13, 20, 22], even if exact complexity results are often elusive due to the intricate nature of the problems (e.g. the exact complexity of solving mean-payoff games is still an open problem). In multi-player non zero-sum games, the *(constrained) existence* of equilibria is also well studied. The existence of simple NEs was established in [12] for mean-payoff and discounted-sum objectives. No decision problem is considered in that paper. The constrained existence of SPEs in quantitative reachability games was proved PSPACE-complete in [8]. We prove here that the complexity is lower when we use NEs to model rationality, as we obtain NP-completeness for the related cooperative synthesis problem. Deciding the constrained existence of SPEs was recently solved for quantitative reachability games in [9] and for mean-payoff games in [5, 6]. The cooperative and non-cooperative rational *synthesis problems* were studied in [25] for games with mean-payoff and discounted-sum objectives when the environment is composed of a single player. The mean-payoff case was proved to be NP-complete and the discounted-sum case was linked to the open target discounted sum problem, which explains the difficulty of solving the problem in this case.

Structure of the Paper. The background is given in Section 2. The formal definitions of the studied problems and our main complexity results are stated in Section 3. The proofs of our results are given for PO rationality in Section 4, and for NE rationality in Section 5. We give a conclusion and future work in Section 6.

2 Background

Arenas and Plays. A (finite) *arena* $\mathcal{A}$ is a tuple $(V, E, \mathcal{P}, (V_i)_{i\in\mathcal{P}})$ where V is a finite set of *vertices*, $E \subseteq V \times V$ is a set of *edges*, $\mathcal{P}$ is a finite set of *players*, and $(V_i)_{i\in\mathcal{P}}$ is a partition of V, where V_i is the set of vertices *owned* by player i. We assume that $v \in V$ has at least one *successor*, i.e., the set $\mathsf{succ}(v) = \{v' \in V \mid (v, v') \in E\}$ is nonempty.

We define a *play* $\pi \in V^\omega$ (resp. a *history* $h \in V^*$) as an infinite (resp. finite) sequence of vertices $\pi_0\pi_1\dots$ such that $(\pi_i, \pi_{i+1}) \in E$ for any two consecutive vertices π_i, π_{i+1}. The *length* $|h|$ of a history h is the number of its vertices. The empty history is denoted ε. Given a play π and two indexes $k < k'$, we write $\pi_{\leq k}$ the prefix $\pi_0 \dots \pi_k$ of π, $\pi_{\geq k}$ the suffix $\pi_k\pi_{k+1}\dots$ of π, and $\pi_{[k,k'[}$ for $\pi_k \dots \pi_{k'-1}$. We denote the first vertex of π by $\mathsf{first}(\pi)$. These notations are naturally adapted to histories. We also write $\mathsf{last}(h)$ for the last vertex of a history $h \neq \varepsilon$. The set of all plays (resp. histories) of an arena $\mathcal{A}$ is denoted $\mathsf{Plays}_\mathcal{A} \subseteq V^\omega$ (resp. $\mathsf{Hist}_\mathcal{A} \subseteq V^*$), and we write Plays (resp. Hist) when the context is clear. For $i \in \mathcal{P}$, the set $\mathsf{Hist}_i \subseteq V^*V_i$ represents all histories ending in a vertex $v \in V_i$. That is, $\mathsf{Hist}_i = \{h \in \mathsf{Hist} \mid h \neq \varepsilon \text{ and } \mathsf{last}(h) \in V_i\}$.

We can *concatenate* two nonempty histories h_1 and h_2 into a single one, denoted $h_1 \cdot h_2$ or h_1h_2 if $(\mathsf{last}(h_1), \mathsf{first}(h_2)) \in E$. When a history can be concatenated to itself, we call it a *cycle*. Furthermore, a play $\pi = \mu\nu\nu\dots = \mu(\nu)^\omega$ where $\mu\nu \in \mathsf{Hist}$ with ν a cycle, is called a *lasso*. The *length* of π is then the length of $\mu\nu$.[2] Given a play π, a *cycle along* π refers to a sequence $\pi_{[m,n[}$ with $\pi_m = \pi_n$. We denote $\mathsf{Occ}(\pi) = \{v \in V \mid \exists n \in \mathbb{N},\ v = \pi_n\}$ the set of all vertices occurring along π, and we say that π *visits* or *reaches* a vertex $v \in \mathsf{Occ}(\pi)$ or a set T such that $T \cap \mathsf{Occ}(\pi) \neq \emptyset$. The previous notions extend to histories.

Given an arena $\mathcal{A}$, if we fix an *initial vertex* $v_0 \in V$, we say that $\mathcal{A}$ is *initialized* and we denote by $\mathsf{Plays}(v_0)$ (resp. $\mathsf{Hist}(v_0)$) all its plays (resp. nonempty histories) starting with v_0. An arena is called *weighted* if it is augmented with a non-negative *weight function* $w_i : E \to \mathbb{N}$ for each player i. We denote by W the greatest weight, i.e., $W = \max\{w_i(e) \mid e \in E, i \in \mathcal{P}\}$. We extend w_i to any history $h = \pi_0 \dots \pi_n$ such that $w_i(h) = \sum_{j=1}^{n} w_i((\pi_{j-1}, \pi_j))$.

[2] To have a well-defined length for a lasso π, we assume that $\pi = \mu(\nu)^\omega$ with $\mu\nu$ of minimal length.

Reachability Games. A *reachability game* is a tuple $\mathcal{G} = (\mathcal{A}, (T_i)_{i\in\mathcal{P}})$ where $\mathcal{A}$ is a weighted arena and $T_i \subseteq V$ is a *target set* for each $i \in \mathcal{P}$. We define a *cost function* $\mathsf{cost}_i : \mathsf{Plays} \to \mathbb{N} \cup \{+\infty\}$ for each player i, such that for all plays $\pi = \pi_0\pi_1 \cdots \in \mathsf{Plays}$, $\mathsf{cost}_i(\pi) = w_i(\pi_0 \dots \pi_n)$ with n the smallest index such that $\pi_n \in T_i$, if it exists and $\mathsf{cost}_i(\pi) = +\infty$ otherwise.

The *reachability objective* of player i is to *minimize* this cost as much as possible, i.e., given two plays π, π' such that $\mathsf{cost}_i(\pi) < \mathsf{cost}_i(\pi')$, player i prefers π to π'. We extend $<$ to tuples of costs as follows: $(\mathsf{cost}_i(\pi))_{i\in\mathcal{P}} < (\mathsf{cost}_i(\pi'))_{i\in\mathcal{P}}$ if $\mathsf{cost}_i(\pi) \le \mathsf{cost}_i(\pi')$ for all $i \in \mathcal{P}$ and there exists some $j \in \mathcal{P}$ such that $\mathsf{cost}_j(\pi) < \mathsf{cost}_j(\pi')$. Given a play π, we denote by $\mathsf{Visit}(\pi)$ the set of players i such that π visits T_i, i.e., $\mathsf{Visit}(\pi) = \{i \in \mathcal{P} \mid \mathsf{cost}_i(\pi) < +\infty\}$. When for all $i \in \mathcal{P}$ and $e \in E$, $w_i(e) = 0$, we speak of *qualitative* reachability games, since $\mathsf{cost}_i(\pi) = 0$ if $\mathsf{Occ}(\pi) \cap T_i \neq \emptyset$ and $+\infty$ otherwise.

Strategies and Mealy Machines. Let $\mathcal{A} = (V, E, \mathcal{P}, (V_i)_{i\in\mathcal{P}})$ be an arena. A *strategy* $\sigma_i : \mathsf{Hist}_i \to V$ for player i maps any history $h \in \mathsf{Hist}_i$ to a vertex $v \in \mathsf{succ}(\mathsf{last}(h))$, which is the next vertex that player i chooses to move to after reaching the last vertex in h. The set of all strategies of player i is denoted Σ_i. A play $\pi = \pi_0\pi_1 \dots$ is *consistent* with σ_i if $\pi_{k+1} = \sigma_i(\pi_0 \dots \pi_k)$ for all $k \in \mathbb{N}$ such that $\pi_k \in V_i$. Consistency is naturally extended to histories. A tuple of strategies $\sigma = (\sigma_i)_{i\in\mathcal{P}}$ with $\sigma_i \in \Sigma_i$, is called a *strategy profile*. In an arena initialized at v_0, we limit the domain of each strategy σ_i to $\mathsf{Hist}_i(v_0)$; the play π starting from v_0 and consistent with each σ_i is denoted $\langle\sigma\rangle_{v_0}$ and called *outcome*.

Given an initialized arena $\mathcal{A}$, we can encode a strategy or a set of strategies by a (finite) *nondeterministic Mealy machine* [7, 19] $\mathcal{M} = (M, m_0, \delta, \tau)$ on $\mathcal{A}$, where M is a finite set of *memory states*, $m_0 \in M$ is the *initial state*, $\delta : M \times V \to 2^M$ is the *update* function, and $\tau : M \times V_i \to 2^V$ is the *next-move* function. Such a machine embeds a (possibly infinite) set of strategies σ_i for player i, called *compatible* strategies. Formally, σ_i is compatible with $\mathcal{M}$ if there exists a mapping $h \mapsto m_h$ such that $m_{hv} \in \delta(m_h, v)$ for every $hv \in \mathsf{Hist}(v_0)$ (with $m_h = m_0$ when h is empty), and when $v \in V_i$, $\sigma_i(hv) \in \tau(m_h, v)$. An example of such a machine $\mathcal{M}$ is given in Appendix A. We denote by $[\![\mathcal{M}]\!]$ the set of all strategies compatible with $\mathcal{M}$. The *memory size* of $\mathcal{M}$ is equal to $|M|$. We say that $\mathcal{M}$ is *deterministic* when the image of both functions δ and τ is a singleton. Thus when $\mathcal{M}$ is deterministic, $[\![\mathcal{M}]\!] = \{\sigma_i\}$ and σ_i is called *finite-memory*, and when additionally $|M| = 1$, σ_i is called *memoryless*.

3 Studied Problems

In this section, within the context of rational synthesis and verification, we consider a reachability game $\mathcal{G} = (\mathcal{A}, (T_i)_{i\in\mathcal{P}})$ with $\mathcal{A}$ an initialized weighted arena and $\mathcal{P} = \{0, 1, \dots, t\}$ such that player 0 is a specific player, often called *system* or *leader*, and the other players $1, \dots, t$ compose the *environment* and are called *followers*. Player 0 announces his strategy σ_0 at the beginning of the game and is not allowed to change it according to the behavior of the other players. The response of those players to σ_0 is supposed to be *rational*, where the rationality can be described as the outcome of a *Nash equilibrium* [35] or as a *Pareto-optimal* play [18].

Nash Equilibria. A strategy profile for the environment is a Nash equilibrium if no player has an incentive to unilaterally deviate from this profile. In other words, no player can improve his cost by switching to a different strategy, assuming that the other players stick to their current strategies. Formally, given the initial vertex v_0 and a strategy σ_0 announced by player 0, a strategy profile $\sigma = (\sigma_i)_{i\in\mathcal{P}}$ is called a 0-*fixed Nash equilibrium* (0-fixed NE) if for every player $i \in \mathcal{P}\backslash\{0\}$ and every strategy $\tau_i \in \Sigma_i$, it holds that $\mathsf{cost}_i(\langle\sigma\rangle_{v_0}) \le \mathsf{cost}_i(\langle\tau_i, \sigma_{-i}\rangle_{v_0})$, where σ_{-i} denotes $(\sigma_j)_{j\in\mathcal{P}\backslash\{i\}}$, i.e., τ_i is not a profitable deviation. We also say that σ is a σ_0-*fixed NE* to emphasize the strategy σ_0 of player 0.

Pareto-Optimality. When all players collaborate to obtain a best cost for everyone, we need another concept of rationality. In that case, we suppose that the players in $\mathcal{P}\setminus\{0\}$ form a *single* player, player 1, that has a *tuple* of targets sets $(T_i)_{i\in\{1,\dots,t\}}$. For each play $\pi \in \mathsf{Plays}(v_0)$, player 1 gets a cost tuple $\mathsf{cost}_{\mathrm{env}}(\pi) = (\mathsf{cost}_i(\pi))_{i\in\{1,\dots,t\}}$, and prefers π to π' if $\mathsf{cost}_{\mathrm{env}}(\pi) < \mathsf{cost}_{\mathrm{env}}(\pi')$ for the componentwise partial order $<$ over $(\mathbb{N} \cup \{+\infty\})^t$. Given such a modified game and a strategy σ_0 announced by player 0, we consider the set C_{σ_0} of cost tuples of plays consistent with σ_0 that are *Pareto-optimal* for player 1, i.e., minimal with respect to $<$. Hence, $C_{\sigma_0} = \min\{\mathsf{cost}_{\mathrm{env}}(\pi) \mid \pi \in \mathsf{Plays}(v_0) \text{ consistent with } \sigma_0\}$. Notice that C_{σ_0} is an antichain. A cost tuple p (called cost in the sequel) is said to be *σ_0-fixed Pareto-optimal* (σ_0-fixed PO or simply 0-fixed PO) if $p \in C_{\sigma_0}$. Similarly, a play is said to be σ_0-fixed PO if its cost is σ_0-fixed PO.

In some problems studied in this paper, we will have to consider games such that all vertices owned by player 0 have only one successor, which means that player 0 has no choice but to choose this successor. In this case, we say that *player* 1 *is the only one to play.*

Rational Verification. We now present the studied decision problems related to the concept of *rational verification.* Given some threshold $c \in \mathbb{N}$, the goal is to verify that a strategy σ_0 announced by player 0 guarantees him a cost $\mathsf{cost}_0(\pi) \leq c$ whatever the rational response π of the environment. By rational response, we mean either a σ_0-fixed NE outcome π, or a σ_0-fixed PO play π. The strategy σ_0 is usually given as a deterministic Mealy machine. We can go further: with a nondeterministic Mealy machine, we want to verify whether all strategies $\sigma_0 \in [\![\mathcal{M}]\!]$ are solutions. In the latter case, we speak about *universal* verification.

▶ **Problem 1.** *Given a reachability game $\mathcal{G}$ with an initialized arena, a nondeterministic Mealy machine $\mathcal{M}_0$ for player 0, and a threshold $c \in \mathbb{N}$,*

- *If $[\![\mathcal{M}_0]\!] = \{\sigma_0\}$, the* Non-Cooperative Nash Verification problem (NCNV) *asks whether for all σ_0-fixed NEs σ, it holds that $\mathsf{cost}_0(\langle\sigma\rangle_{v_0}) \leq c$.*
- *The* Universal Non-Cooperative Nash Verification problem (UNCNV) *asks whether for all $\sigma_0 \in [\![\mathcal{M}_0]\!]$ and all σ_0-fixed NEs σ, it holds that $\mathsf{cost}_0(\langle\sigma\rangle_{v_0}) \leq c$.*
- *If $[\![\mathcal{M}_0]\!] = \{\sigma_0\}$, the* Non-Cooperative Pareto Verification problem (NCPV) *asks whether for all σ_0-fixed PO plays π, it holds that $\mathsf{cost}_0(\pi) \leq c$.*
- *The* Universal Non-Cooperative Pareto Verification problem (UNCPV) *asks whether for all $\sigma_0 \in [\![\mathcal{M}_0]\!]$ and all σ_0-fixed PO plays π, it holds that $\mathsf{cost}_0(\pi) \leq c$.*

Rational Synthesis. We consider the more challenging problem of *rational synthesis.* Given a threshold $c \in \mathbb{N}$, the goal is to synthesize a strategy σ_0 for player 0 (instead of verifying some σ_0) that guarantees him a cost $\mathsf{cost}_0(\pi) \leq c$ whatever the rational response π of the environment. We also consider the simpler problem where the environment *cooperates* with the leader by proposing *some* rational response π that guarantees him a cost $\mathsf{cost}_0(\pi) \leq c$.

▶ **Problem 2.** *Given a reachability game $\mathcal{G}$ with an initialized arena and a threshold $c \in \mathbb{N}$,*

- *The* Cooperative Nash Synthesis (CNS) *problem asks whether there exists $\sigma_0 \in \Sigma_0$ and a σ_0-fixed NE σ such that $\mathsf{cost}_0(\langle\sigma\rangle_{v_0}) \leq c$.*
- *The* Non-Cooperative Nash Synthesis (NCNS) *problem asks whether there exists $\sigma_0 \in \Sigma_0$ such that for all σ_0-fixed NEs σ, it holds that $\mathsf{cost}_0(\langle\sigma\rangle_{v_0}) \leq c$.*
- *The* Cooperative Pareto Synthesis (CPS) *problem asks whether there exists $\sigma_0 \in \Sigma_0$ and a σ_0-fixed PO play π such that $\mathsf{cost}_0(\pi) \leq c$.*
- *The* Non-Cooperative Pareto Synthesis (NCPS) *problem asks whether there exists $\sigma_0 \in \Sigma_0$ such that for all σ_0-fixed PO plays π, it holds that $\mathsf{cost}_0(\pi) \leq c$.*

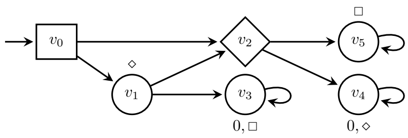

Figure 1 An example illustrating the two concepts of rational response.

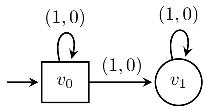

Figure 2 An example showing that PO lasso plays in the coNCPV problem may have an exponential length.

▶ **Example 3.** To illustrate these problems, let us study a simple example depicted in Figure 1 with three players: the system, player 0, and two players in the environment, players $\square$ and $\diamond$. Player 0 owns the circle vertices, player $\square$ owns the square initial vertex v_0, and player $\diamond$ owns the diamond vertex v_2. Each player i has a target set, $T_0 = \{v_3, v_4\}$, $T_\square = \{v_3, v_5\}$ and $T_\diamond = \{v_1, v_4\}$, and a constant weight $w_i(e) = 1$ for all $e \in E$. When a vertex v is in T_i, we depict the symbol of player i nearby v. As the graph is acyclic, the possible player strategies are all memoryless. In the sequel, we thus only indicate the successor chosen by the player.

Let us show that σ_0 defined by $\sigma_0(v_1) = v_2$ is a solution to the NCNS problem with the threshold $c = 3$. Given σ_0, there exist four distinct strategy profiles $\sigma = (\sigma_0, \sigma_\square, \sigma_\diamond)$. When, for example, $\sigma_\square(v_0) = v_2$ and $\sigma_\diamond(v_2) = v_5$, we abusively denote σ as $\{v_0 \to v_2, v_2 \to v_5\}$:

- $\{v_0 \to v_2, v_2 \to v_5\}$ is not a σ_0-fixed NE because its outcome $\pi_1 = v_0v_2(v_5)^\omega$ has a infinite cost for player $\diamond$ who will deviate from v_2 to v_4 to get a cost of 2;
- similarly, $\{v_0 \to v_1, v_2 \to v_5\}$ with outcome $\pi_2 = v_0v_1v_2(v_5)^\omega$ is not a σ_0-fixed NE;
- the profile $\{v_0 \to v_1, v_2 \to v_4\}$ is a σ_0-fixed NE, its outcome is $\pi_3 = v_0v_1v_2(v_4)^\omega$ with $\mathsf{cost}_\square(\pi_3) = +\infty$, $\mathsf{cost}_\diamond(\pi_3) = 1$ and $\mathsf{cost}_0(\pi_3) = 3 \leq c$, so if player $\square$ deviates from v_1 to v_2, his cost is still $+\infty$, and player $\diamond$ has no incentive to deviate since $\mathsf{cost}_\diamond(\pi_3)$ is already the smallest available;
- the profile $\{v_0 \to v_2, v_2 \to v_4\}$ with the outcome $\pi_4 = v_0v_2(v_4)^\omega$ is also a σ_0-fixed NE and $\mathsf{cost}_0(\pi_4) = 2 \leq c$.

So, σ_0 is a solution to the NCNS problem with $c = 3$, but not with $c = 2$. It is also a solution for the CNS problem. One can verify that σ_0' such that $\sigma_0'(v_1) = v_3$ is a solution to the NCNS problem with $c = 2$, since the only σ_0'-fixed NE outcome is $\pi_5 = v_0v_1(v_3)^\omega$.

We now show that σ_0 is not a solution to the CPS problem with $c = 2$. Let us consider the same four outcomes as before. Their cost for the environment are: $\mathsf{cost}_{\mathrm{env}}(\pi_1) = (2, +\infty)$, $\mathsf{cost}_{\mathrm{env}}(\pi_2) = (3, 1)$, $\mathsf{cost}_{\mathrm{env}}(\pi_3) = (+\infty, 1)$, and $\mathsf{cost}_{\mathrm{env}}(\pi_4) = (+\infty, 2)$, meaning that $C_{\sigma_0} = \{(2, +\infty), (3, 1)\}$. Consequently, the only σ_0-fixed PO plays are π_1 and π_2, both giving a cost of $+\infty$ to player 0. However, the strategy σ_0' is a solution, as there is only one σ_0'-fixed PO play, $\pi_5 = v_0v_1(v_3)^\omega$, with $\mathsf{cost}_{\mathrm{env}}(\pi_5) = (2, 1)$ and $\mathsf{cost}_0(\pi_5) = 2$.

Main Results. Our main results for Problems 1-2 are the following ones when the rational responses of the environment are 0-fixed PO plays. One problem was already solved in [11].

▶ **Theorem 4.**
(a) *The Non-Cooperative Pareto Verification problem is* Π_2^{P}*-complete.*
(b) *The Universal Non-Cooperative Pareto Verification problem is* PSPACE*-complete.*
(c) *The Cooperative Pareto Synthesis problem is* PSPACE*-complete.*
(d) *The Non-Cooperative Pareto Synthesis problem is* NEXPTIME*-complete [11].*

For 0-fixed NE responses of the environment, we obtain the next main results.

▶ **Theorem 5.**
(a) *The Non-Cooperative Nash Verification problem is* coNP*-complete.*
(b) *The Universal Non-Cooperative Nash Verification problem is* coNP*-complete.*
(c) *The Cooperative Nash Synthesis problem is* NP*-complete.*
(d) *The Non-Cooperative Nash Synthesis problem is* EXPTIME*-hard, already with a two-player environment. With a one-player environment, it is in* EXPTIME *and* PSPACE*-hard.*

These complexity results depend on the size $|V|$ of the arena, the number t of players i (resp. target sets T_i) in case of 0-fixed NE responses (resp. 0-fixed PO responses), the maximal weight W encoded in binary appearing in the functions w_i, the threshold c encoded in binary, and the size $|M|$ of the Mealy machine $\mathcal{M}_0$ (for the verification problems). Note that for all problems except the NCNS problem, the complexity classes are the same for both qualitative and quantitative frameworks (see Table 1). Hence, in the case of a *unary* encoding of the weights and the threshold c, we get the same complexity classes. Due to space constraints, only the most challenging proofs are provided in the paper, while the other proofs or results derived from the literature are deferred in the long version of this paper [17].

In this paper, we focus on zero or positive weights, because with negative weights, there are simple examples of one-player games with no NE or no PO plays (thus with no rational responses). Furthermore, considering any weights leads to the undecidability of the NCNS and NCPS problems. Those results are obtained by reduction from the undecidability of zero-sum multidimensional shortest path games [40, 41]. See details in the long version of this paper [17].

▶ **Theorem 6.** *With integer weight functions, the Non-Cooperative Nash Synthesis problem and the Non-Cooperative Pareto Synthesis problem are undecidable.*

4 Pareto-Optimality

In this section, we provide the proofs of the upper bounds in Theorem 4. Recall that the environment is here composed of the sole player 1 having t target sets T_i, and his rational responses to a strategy σ_0 announced by player 0 are σ_0-fixed PO plays. The lower bounds are proved in the long version [17] with reductions from QBF or some of its variants [42]. All those reductions already hold for qualitative reachability games. We thus obtain the same complexity classes as in Theorem 4 for this class of games, as indicated in Table 1.

To solve the two verification problems (NCPV and UNCPV), we first construct the product game[3] $\mathcal{G} \times \mathcal{M}_0$ of size polynomial in $\mathcal{G}$ and $\mathcal{M}_0$, and we *assume* to directly work with this game, *again denoted* $\mathcal{G}$. Note that in the product game, when $\mathcal{M}_0$ is nondeterministic, player 0 is able to play any strategy σ_0 compatible with $\mathcal{M}_0$, and when $\mathcal{M}_0$ is deterministic, the verification problems are simplified as there is a single compatible strategy σ_0. The complement of the (U)NCPV problem has many similarities with the CPS problem:

▶ **Problem 7.** *The* complement of the (U)NCPV problem (co(U)NCPV) *asks whether there exists $\sigma_0 \in \Sigma_0$ and a σ_0-fixed PO play π such that* $\mathsf{cost}_0(\pi) > c$.

Indeed, the statement is the same except that the inequality $\mathsf{cost}_0(\pi) \leq c$ in the CPS problem is here replaced by $\mathsf{cost}_0(\pi) > c$. To prove the upper bounds of Theorem 4, we thus have to solve the decision problem "do there exist $\sigma_0 \in \Sigma_0$ and a σ_0-fixed PO play π such that $\mathsf{cost}_0(\pi) \sim c$?" with $\sim \in \{\leq, >\}$. In short, the algorithm to solve the CPS problem and the complement of the (U)NCPV problem proceeds through the following steps:

[3] The product of a game with a Mealy machine is recalled in Appendix A.

1. Guess a play π in the form $\pi = \mu(\nu)^\omega$ in polynomial time. The length of the lasso is polynomial or exponential, depending on the studied problem. In the latter case, we will guess a succinct representation of the lasso by using Parikh automata [23, 32].
2. Compute in polynomial time $\mathsf{cost}_{\mathrm{env}}(\pi)$ and verify in polynomial time that $\mathsf{cost}_0(\pi) \sim c$.
3. Verify that player 0 has a strategy σ_0, with π consistent with σ_0, that guarantees that $\mathsf{cost}_{\mathrm{env}}(\pi)$ is σ_0-fixed PO. This last step will be done in coNP or in PSPACE, depending on the studied problem.

Therefore, if a strategy σ_0 exists as in Step 3, the σ_0-fixed PO play π such that $\mathsf{cost}_0(\pi) \sim c$ is the lasso of Step 1. Let us now provide detailed proofs for these three steps.

4.1 Existence of Lassos

The goal is this section is to prove the next lemma stating that one can always suppose that π is a lasso. For that purpose, we use a classical approach consisting of removing cycles [10, 14, 21].

▶ **Lemma 8.** *Let $\sigma_0 \in \Sigma_0$ and π be a σ_0-fixed PO play π such that $\mathsf{cost}_0(\pi) \sim c$. Then there exist $\sigma_0' \in \Sigma_0$ and a σ_0'-fixed PO play $\pi' = \mu(\nu)^\omega$ such that $\mathsf{cost}_0(\pi') \sim c$. Moreover, $\mathsf{Visit}(\mu) = \mathsf{Visit}(\mu\nu)$ and*

- *if $\mathsf{cost}_0(\pi) \leq c$, then $|\mu| \leq (t+1)|V|$, $|\nu| \leq |V|$, $\mathsf{cost}_{\mathrm{env}}(\pi') \in \{0, 1, \dots, B, +\infty\}^t$, with $B = (t+2)|V|W$,*
- *if $\mathsf{cost}_0(\pi) > c$, then $|\mu| \leq c + (t+1)|V|$, $|\nu| \leq |V|$, $\mathsf{cost}_{\mathrm{env}}(\pi') \in \{0, 1, \dots, B, +\infty\}^t$, with $B = (c + (t+2)|V|)W$.*

Proof. Let $\pi = \pi_0\pi_1 \dots$ be a σ_0-fixed PO play such that $\mathsf{cost}_0(\pi) \sim c$.

Suppose that $\mathsf{cost}_0(\pi) \leq c$. Consider, along π, any two consecutive first visits to two target sets, say T_i and T_j. If there exists $m < n$ such that $\pi_n = \pi_m$ between these two visits, we remove the cycle $\pi_{[m,n[}$ from π. We repeat this process until there are less than $|V|$ vertices between the two visits, for any such pair T_i, T_j, but also between π_0 and the first visit to a target set. Let us denote π' the resulting play. Consider now along π' the last first visit to a target set, say at index k. We then seek for the first repeated vertex $\pi'_{\ell_1} = \pi'_{\ell_2}$ with $k \leq \ell_1 < \ell_2$ after k. In this way, we obtain $\nu = \pi'_{[\ell_1,\ell_2[}$ with $|\nu| \leq |V|$ and $\mu = \pi'_{[0,\ell_1[}$ with $|\mu| \leq (t+1)|V|$. So, we get the required lasso $\mu(\nu)^\omega$ such that $\mathsf{Visit}(\mu) = \mathsf{Visit}(\mu\nu)$, $\mathsf{cost}_0(\mu(\nu)^\omega) \leq \mathsf{cost}_0(\pi) \leq c$, and $\mathsf{cost}_{\mathrm{env}}(\mu(\nu)^\omega) \in \{0, 1, \dots, B, +\infty\}^t$, with $B = (t+2)|V|W$.

The case $\mathsf{cost}_0(\pi) > c$ is treated similarly, except that we cannot remove cycles along the longest prefix h of π such that $\mathsf{cost}_0(h) \leq c$, as this operation might decrease the cost of player 0. We thus get $|\mu| \leq c + (t+1)|V|$, $\mathsf{cost}_0(\mu(\nu)^\omega) > c$, and $\mathsf{cost}_{\mathrm{env}}(\mu(\nu)^\omega) \in \{0, 1, \dots, B, +\infty\}^t$, with $B = (c + (t+2)|V|)W$.

It remains to explain how to construct a strategy σ_0' from σ_0 such that $\pi' = \mu(\nu)^\omega$ is σ_0'-fixed PO. First, σ_0' is built in a way to produce π'. Second, we have to define σ_0' outside π', i.e., from any $h'v$, with $v \in V$, such that h' is prefix of π' but not $h'v$. Let h be such that the elimination of cycles done in π, restricted to h, leads to h'. We then define $\sigma_0'(h'g) = \sigma_0(hg)$ for all histories $g \in \mathsf{Hist}(v)$. Notice that σ_0' is the required strategy as the elimination of cycles in a history or a play decreases the costs. ◀

▶ **Example 9.** When $\mathsf{cost}_0(\pi) > c$, Lemma 8 provides a bound on $|\mu\nu|$ that is exponential in the binary encoding of c. In Figure 2, we present a small example of a reachability game showing that this is unavoidable. The initial vertex v_0 is owned by player 1, v_1 is owned by player 0, and there are two weight functions w_0 and w_1 (thus $t = 1$). Both players have the same target set: $T_0 = T_1 = \{v_1\}$. Notice that player 1 is the only one to play, and a play

$\pi \in \mathsf{Plays}(v_0)$ is PO if and only if visits T_1 (and has $\mathsf{cost}_{\mathrm{env}}(\pi) = 0$). Hence, given a threshold c, any PO play π with $\mathsf{cost}_0(\pi) > c$ is equal to $v_0^k(v_1)^\omega$ with $k > c$. The length $|v_0^k v_1|$ is thus greater than c. Therefore, Step 1 of our decision algorithm for the co(U)NCPV cannot guess an explicit representation $\mu(\nu)^\omega$ if we want to stick to a polynomial time algorithm.

4.2 Particular Zero-sum Games

Now that we know we can limit our study to lassos π, Step 3 requires to verify that player 0 has a strategy σ_0 ensuring that $\mathsf{cost}_{\mathrm{env}}(\pi)$ is σ_0-fixed PO. Before going deeper into this step, we need to study some particular two-player zero-sum games.[4] Let $\mathcal{A} = (V, E, \mathcal{P}, (V_i)_{i\in\mathcal{P}}, (w_i)_{i\in\{1,\dots,t\}})$ be an arena with $\mathcal{P} = \{Eve, Adam\}$ and equipped with t weight functions $w_i : E \to \mathbb{N}$. We suppose that $\mathcal{A}$ is initialized with $v_0 \in V$. We fix t target sets $T_i \subseteq V$ and t constants $d_i \in \mathbb{N}^{>0} \cup \{+\infty\}$. We denote by $\mathcal{G} = (\mathcal{A}, \Omega)$ a zero-sum game whose *objective* Ω is a Boolean combination of the following objectives:

- $\mathsf{Reach}_{<d_i}(T_i) = \{\pi \in \mathsf{Plays}(v_0) \mid \mathsf{cost}_i(\pi) < d_i\}$ called *bounded reachability objective*, and
- $\mathsf{Safe}_{\geq d_i}(T_i) = \mathsf{Plays}(v_0) \setminus \mathsf{Reach}_{<d_i}(T_i)$ called *bounded safety objective*.

Solving such a game $\mathcal{G}$ means to decide whether *Eve* has a strategy σ such that all plays $\pi \in \mathsf{Plays}(v_0)$ consistent with σ belong to the objective Ω. If such a strategy σ exists, we say that σ is *winning for* Ω and that the initial vertex v_0 *is winning for* Eve *for* Ω.

For the PO-check required for Step 3, will see in Section 4.3 that we need to solve the zero-sum games stated in the next two propositions, where the constants d_i are encoded in binary. The second proposition will be used in the general case of nondeterministic Mealy machines $\mathcal{M}_0$ while the first one will be used in the deterministic case. Proposition 10 is a quantitative extension of a result in [24] about (qualitative) generalized reachability games.

▶ **Proposition 10.** *Let $\mathcal{G} = (\mathcal{A}, \Omega)$ be a zero-sum game with $\Omega = \bigcap_{1\leq i\leq t} \mathsf{Reach}_{<d_i}(T_i)$ and* Eve *is the only one to play. Deciding whether v_0 is winning for* Eve *is an* NP*-complete problem.*

Proof. We first notice that if *Eve* has a winning strategy from v_0, i.e., there exists a play $\pi \in \Omega$, then we can eliminate cycles as in the proof of Lemma 8. Therefore, there exists a lasso $\pi' = \mu(\nu)^\omega \in \Omega$ where $|\mu\nu| \leq (t+2)|V|$. Thus, to get an algorithm in NP, we guess such a lasso π' and verify that $\mathsf{cost}_i(\pi') < d_i$ for each $i \in \{1, \dots, t\}$. This is possible in polynomial time with the costs encoded in binary. It is proved in [24] that solving (qualitative) generalized reachability games with $V_{Adam} = \emptyset$ is NP-complete. Our problem is thus NP-hard by a reduction from the previous problem with the same arena, the weight functions assigning a null weight to all edges, and by setting $(d_1, \dots, d_t) = (+\infty, \dots, +\infty)$. ◀

The next proposition, of potential independent interest, is easily extended to any positive Boolean combinations of bounded safety objectives.

▶ **Proposition 11.** *Let $\mathcal{G} = (\mathcal{A}, \Omega)$ be a zero-sum game where $\Omega = \Omega^{(1)} \cup \Omega^{(2)}$, with $\Omega^{(1)} = \left(\bigcap_{1\leq i\leq t} \mathsf{Safe}_{\geq d_i}(T_i)\right)$ and $\Omega^{(2)} = \left(\bigcup_{1\leq i\leq t} \mathsf{Safe}_{\geq d_i+1}(T_i)\right)$, and such that $+\infty + 1 = +\infty$. Then, deciding whether v_0 is winning for* Eve *is in* PSPACE*.*

Proof. We solve the game $(\mathcal{A}, \Omega)$ by using a recursive algorithm. To know whether v_0 is winning for *Eve*, we run a depth-first search over a finite tree rooted at v_0 that is the (truncated) unraveling of $\mathcal{A}$, and we keep track of the accumulated weights along the explored

[4] We suppose that the reader is familiar with this concept.

branch as a tuple $(c_i)_{1 \leq i \leq t}$, where each c_i is encoded in binary. Each explored branch h will have its leaf decorated by a boolean $f(h) = \bot$ (*Eve* is losing) or $f(h) = \top$ (*Eve* is winning) according to some rules that we describe below. Then the depth-first search algorithm backwardly assigns a boolean to the internal nodes of the tree according to the following rule: for any $hv \in V^*V_{Eve}$, we have $f(hv) = \top$ if there exists $v' \in \mathsf{succ}(v)$ such that $f(hvv') = \top$, otherwise $f(hv) = \bot$, while for any $hv \in V^*V_{Adam}$, we have $f(hv) = \top$ if for all $v' \in \mathsf{succ}(v)$, $f(hvv') = \top$, otherwise $f(hv) = \bot$. To have an algorithm executing in polynomial space, the depth of the tree must be polynomial.

Along a branch, the rules are the following to stop the exploration (the objective Ω may be modified during the exploration):

- If for some i, the current weight c_i is such that $c_i \geq d_i + 1$ and T_i was not visited, then we can stop the branch h and set $f(h) = \top$. Indeed, $\Omega^{(2)}$ is satisfied, and thus also Ω.
- If for some i, we have $c_i < d_i$ while visiting T_i, then $\Omega^{(1)}$ is not satisfiable anymore, and we continue the exploration with the sole objective $\Omega^{(2)}$ where the i-th objective $\mathsf{Safe}_{\geq d_i+1}(T_i)$ being ignored (as it is not satisfied).
- If for some i, we have $c_i = d_i$ while visiting T_i, then we continue the exploration with Ω such that $\mathsf{Safe}_{\geq d_i}(T_i)$ is removed from $\Omega^{(1)}$ (as it is satisfied) and $\mathsf{Safe}_{\geq d_i+1}(T_i)$ is removed from $\Omega^{(2)}$ (as it is not satisfied).
- If $\Omega^{(1)}$ becomes an empty intersection, then we stop the branch h and set $f(h) = \top$.
- If $\Omega^{(1)}$ has been removed from Ω (because it was not satisfiable anymore) and $\Omega^{(2)}$ becomes an empty union, then we stop the branch h and set $f(h) = \bot$.
- There is one more case to stop the branch h: when some vertex v is visited twice, i.e., $h = gvg'v$ for some $g, g' \in V^*$. Then we stop the branch and set $f(h) = \top$. Indeed, we stand in a better situation in $gvg'v$ than in gv concerning the accumulated weights, as we consider bounded safety objectives.

The last case happens as soon as the explored branch has length $|V| + 1$ and the other cases do not occur. Therefore, as there are t bounded safety objectives in both $\Omega^{(1)}$ and $\Omega^{(2)}$, any branch has a length polynomially bounded by $t|V|$. Moreover, the accumulated weights c_i are all bounded by $t|V|W$, thus stored in a polynomial space when encoded in binary. We can thus decide in polynomial space whether v_0 is winning for *Eve* for Ω. ◀

4.3 Pareto-Optimality Check

Let us come back to our reachability games. We can now solve Step 3 where given a lasso π with $\mathsf{cost}_{\mathrm{env}}(\pi) \in \{0, 1, \ldots, B, +\infty\}^t$ (by Lemma 8), we want to verify whether player 0 has a strategy σ_0 guaranteeing that $\mathsf{cost}_{\mathrm{env}}(\pi)$ is σ_0-fixed PO. If player 1 is the only one to play in the game, it reduces to verify that $\mathsf{cost}_{\mathrm{env}}(\pi)$ is PO. The latter problem is in coNP as stated in the next lemma, while the former is in PSPACE as stated in Lemma 13.

▶ **Lemma 12.** *Suppose that player 1 is the only one to play. Deciding whether a given cost $p \in \{0, 1, \ldots, B, +\infty\}^t$ is PO is in* coNP.

Proof. The cost p is *not* PO if there exists a play $\pi' \in \mathsf{Plays}(v_0)$ such that $\mathsf{cost}_i(\pi') \leq p_i$ for all $i \in \{1, \ldots, t\}$ and $\mathsf{cost}_j(\pi') < p_j$ for some $j \in \{1, \ldots, t\}$. That is, if for some j, there exists a play $\pi' \in \Omega^{(j)} = \bigcap_{i \neq j} \mathsf{Reach}_{<p_i+1}(T_i) \cap \mathsf{Reach}_{<p_j}(T_j)$. Solving the zero-sum game $(\mathcal{A}, \Omega)$ is in NP by Proposition 10. This concludes the proof. ◀

▶ **Lemma 13.** *Given $p = \mathsf{cost}_{\mathrm{env}}(\pi) \in \{0, 1, \ldots, B, +\infty\}^t$ being the cost of a play π, deciding whether player 0 has a strategy σ_0 ensuring that p is σ_0-fixed PO is in* PSPACE.

Proof. To prove the lemma, we first fix a prefix h of π, with $v \in V$, such that hv is not a prefix of π (hv is called a *deviation*), and we study the zero-sum game $(\mathcal{A}, \Omega^{(hv)})$ with the objective $\Omega^{(hv)}$ equal to $\{\pi' \in \mathsf{Plays}(v) \mid \neg(\mathsf{cost}_{\mathrm{env}}(h\pi') < p)\}$. Let us show that deciding whether v is winning for player 0 for $\Omega^{(hv)}$ is in PSPACE. Notice that for each $i \in \{1, \ldots, t\}$ such that h does not visit T_i, we have, with $q_i = w_i(hv)$ and $+\infty - q_i = +\infty$: $\mathsf{cost}_i(h\pi') < p_i$ if and only if $\mathsf{cost}_i(\pi') < p_i - q_i$. Let us rewrite the condition $\neg(p' < p)$ with $p, p' \in \mathbb{N}^t$ as follows: $(\forall i \in \{1, \ldots, t\}\ p'_i \geq p_i) \vee (\exists i \in \{1, \ldots, t\}\ p'_i > p_i)$. Hence, the objective $\Omega^{(hv)}$ can be rewritten as $\left(\bigcap_{\substack{1 \leq i \leq t \\ \mathsf{Occ}(h) \cap T_i = \emptyset}} \mathsf{Safe}_{\geq p_i - q_i}(T_i)\right) \cup \left(\bigcup_{\substack{1 \leq i \leq t \\ \mathsf{Occ}(h) \cap T_i = \emptyset}} \mathsf{Safe}_{\geq p_i - q_i + 1}(T_i)\right)$. By Proposition 11, given the constants p_i and q_i, we can check whether v is winning for player 0 in polynomial space. Notice that each q_i can be computed in polynomial space by accumulating the weights, with respect to w_i, as long as T_i is not visited (as $q_i \leq p_i$).

Second, given two deviations $hv, h'v$ ending with the same vertex v and such that h is prefix of h', if $\mathsf{Visit}(h') = \mathsf{Visit}(h)$ and v is winning for $\Omega^{(hv)}$, then v is also winning for $\Omega^{(h'v)}$ (with the same strategy). Indeed, the constants q'_i for $h'v$ are greater than the constants q_i for hv. We are thus in a "better situation" than in $\Omega^{(h'v)}$. So, it suffices to consider polynomially many deviations hv, as π can visit at most t target sets and there are at most $|V|$ vertices v.

Finally, deciding whether player 0 has a strategy σ_0 ensuring that p is σ_0-fixed PO amounts to solving the zero-sum games $(\mathcal{A}, \Omega^{(hv)})$ for polynomially many deviations hv. If player 0 has a winning strategy σ_{hv} for all those games, the required strategy σ_0 is defined as $\sigma_0(g) = \sigma_{hv}(vg')$ for all histories g such that $g = hvg'$ with the longest prefix h of π. ◀

4.4 Upper Bounds

We are now ready to prove the upper bounds in Theorem 4 by providing the announced algorithms for Steps 1-3. The proof is divided according to the considered problem. We need to recall [23] that a Parikh automaton is a nondeterministic finite automaton (NFA) over an alphabet Σ and whose transitions are weighted by tuples in $\mathbb{N}^k$, together with a semilinear set $\mathbf{C} \subseteq \mathbb{N}^k$. It accepts a word $w \in \Sigma^*$ if there exists a run on w ending on an accepting state such that the sum of all encountered weight tuples belongs to $\mathbf{C}$. The non-emptiness problem for Parikh automata is NP-complete for numbers encoded in binary [23].

Proof of the upper bounds in Theorem 4. We begin with the CPS problem (Theorem 4.c). Let us give an algorithm in PSPACE that decides whether there exist $\sigma_0 \in \Sigma_0$ and a σ_0-fixed PO play π such that $\mathsf{cost}_0(\pi) \leq c$. By Lemma 8, we guess a lasso $\pi = \mu(\nu)^\omega$ with $|\mu\nu| \leq (t+2)|V|$, in time polynomial in $|V|$ and t. Then, we compute $p = \mathsf{cost}_{\mathrm{env}}(\pi)$ and $\mathsf{cost}_0(\pi)$ and check whether $\mathsf{cost}_0(\pi) \leq c$. This can be done in time polynomial in t, $|V|$, and the binary encoding of W and c by Lemma 8. Finally, by Lemma 13, we verify in polynomial space whether player 0 has a strategy σ_0 ensuring that p is σ_0-fixed PO.

For the NCPV problem (Theorem 4.a), recall that we consider its complementary coNCPV problem (see Problem 7), and that player 1 is the only one to play. We begin by giving an algorithm in NP for Step 1 and 2. Lemma 8 does not provide a polynomial bound on the length of the lasso $\pi = \mu(\nu)^\omega$ due to the threshold c given in binary. However, we will guess a succinct representation of π by using Parikh automata.

The idea is the following one. Along the prefix μ of the lasso π, some target sets $T_{k_1}, \ldots, T_{k_n}$ are visited, with $n \leq t$, such that the first visits are in vertices $\pi_{\ell_1}, \ldots, \pi_{\ell_n}$ with $\ell_1 < \cdots < \ell_n$. And after μ, no more target sets are visited along $\mu\nu$ (see Lemma 8). We start by guessing a sequence $v_0, v_1, \ldots, v_n, v_{n+1}$ of vertices, called *markers*, with the aim that v_0 is the initial vertex, $v_i = \pi_{\ell_i}$, $1 \leq i \leq n$, and $v_{n+1} = \mathsf{first}(\nu)$. By Lemma 8, we know that $\mathsf{cost}_{\mathrm{env}}(\pi) \in \{0, 1, \dots, B, +\infty\}^t$, where $B = (c + (t + 2)|V|)W$. We thus guess a tuple $(p_0, p_1, \dots, p_t) \in \{0, 1, \dots, B, +\infty\}^t$ with the aim that $(p_1, \dots, p_t) = \mathsf{cost}_{\mathrm{env}}(\mu)$ and $p_0 = w_0(\mu)$. We also guess for each *portion* $\pi_{[\ell_i,\ell_{i+1}]}$, $i \leq n$,

- a weight $q_0^{(i)} \in \{0, 1, \dots, B\}$ for player 0 with the aim that $q_0^{(i)} = w_0(\pi_{[\ell_i,\ell_{i+1}]})$ and $w_0(\mu) = p_0 = \Sigma_i q_0^{(i)}$,
- a "useful" environment weight tuple, i.e., for all $j \in \{1, \dots, t\}$, a weight $q_j^{(i)} \in \{0, 1, \dots, B\}$ such that $\pi_{[0,\ell_i]}$ does not visit T_j, with the aim that $q_j^{(i)} = w_j(\pi_{[\ell_i,\ell_{i+1}]})$ and $\mathsf{cost}_j(\mu) = p_j = \Sigma_i q_j^{(i)}$.[5]

We can guess in polynomial time the sequence $v_0, v_1, \dots, v_n, v_{n+1}$ and the constants p_j, $q_j^{(i)}$ encoded in binary, as $n \leq t$ and $B = (c + (t + 2)|V|)W$. We then check in polynomial time that v_0 is the initial state, that each v_i belongs to a distinct target set T_{k_i}, $1 \leq i \leq n$, that $p_j = \Sigma_i q_j^{(i)}$ for each j, and that $p_0 > c$ for the given threshold c.[6]

It remains to check the existence of polynomially many paths:

- For each $i \leq n$, the existence of a path $\rho^{(i)}$ from v_i to v_{i+1} on a subgraph of $\mathcal{A}$ restricted to some sets $V^{(i)}$ and $E^{(i)}$ of vertices and edges respectively, and to some weight functions, such that $w_j(\rho^{(i)}) = q_j^{(i)}$ for all j.
- The existence of a path from v_{n+1} to itself (the cycle ν) that visits no new target set with respect to T_{k_i}, $1 \leq i \leq n$.

The first check can be done thanks to Parikh automata : one can decide in NP the existence of a path in a subgraph of $\mathcal{A}$ between two given vertices and with a given weight tuple $\bar{q}$ (the subgraph is seen as a Parikh automaton with $\Sigma = \{\#\}$ and $\mathbf{C} = \{\bar{q}\}$).[7] The set $V^{(i)}$ is defined as $V \setminus \left(\bigcup_{j>i+1} T_{k_j} \cup \bigcup_{p_j=+\infty} T_j\right)$, and the set $E^{(i)}$ as $(E \cap V^{(i)} \times V^{(i)}) \setminus \{(v, v') \mid v \in T_{k_{i+1}}\}$. Indeed, for the portion $\pi_{[\ell_i,\ell_{i+1}]}$, we do not allow to prematurely visit a target set T_{k_j}, $j \geq i + 1$, except $v_{i+1} \in T_{k_{i+1}}$, and there are target sets that we do not want to visit at all. We also remove the weight function w_{k_j} with $j \in \{1, \dots, i\}$. The second check can be done thanks to classical automata, by restricting the set of vertices to $V \setminus \left(\bigcup_{p_j=+\infty} T_j\right)$. To show that the coNCPV problem is in Σ_2^{P}, in the previous algorithm in NP that guesses a lasso π with $\mathsf{cost}_{\mathrm{env}}(\pi) = p$, we add an oracle in coNP to check whether p is a PO cost thanks to Lemma 12. As $\mathsf{NP}^{\mathsf{coNP}} = \Sigma_2^{\mathsf{P}}$, we get that the NCPV problem is in Π_2^{P}.

It remains to show that the coUNCPV problem is in PSPACE to get the upper bound of Theorem 4.b). The approach is to guess a cost $p \in \{0, \dots, B, +\infty\}^t$ and a length ℓ for the exponential lasso π of Lemma 8, whose both encodings in binary use a polynomial space. We guess π vertex by vertex, by only storing the current edge (u, u'), the current accumulated weight $(c_0, c_1, \dots, c_t)$ on each dimension, and which target sets T_i have already been visited. At any time, the stored information uses a polynomial space. At each guess, we apply the reasoning of Lemma 13 to check in polynomial space whether player 0 can ensure that p is a PO cost from each vertex $v \neq u'$ successor of u (i.e., from any deviation of π). We also check that for each first visit to a target set T_i, we have $c_i = p_i$ if $i \in \{1, \dots, t\}$, and $c_i > c$ if $i = 0$. At each guess, a counter is incremented until reaching the length ℓ, where we stop guessing π and finally check whether $p_i = +\infty$ for each T_i that has not been visited.

This completes the proof as Theorem 4.d is established in [11]. ◀

[5] If $\pi_{[0,\ell_i]}$ visits T_j, then $\mathsf{cost}_j(\pi)$ is already known as $\mathsf{cost}_j(\pi) = \mathsf{cost}_j(\pi_{[0,\ell_i]})$.

[6] To keep the proof readable, we assume that each v_i belongs to one target set T_{k_i}. In general, it could belong to several target sets. The proof is easily adapted by considering the union of target sets.

[7] We do not need to use an oracle here. It suffices to plug the NP algorithm for Parikh automata in ours as if the required path exists, our algorithm will find it in polynomial time.

5 Nash Equilibria

We now discuss the proofs of Theorem 5. The environment is here composed of t players whose rational responses to a strategy σ_0 of player 0 are σ_0-fixed NE outcomes.

The upper bounds for (U)NCNV and CNS problems given in Theorem 5.a-c are proved with the same approach as for Pareto optimality, limited to Steps 1-2. There is no need for Step 3, thanks to a well-known characterization of NE outcomes (based on the values of some two-player zero-sum games, see e.g. [10, 16] or the long version of this paper [17]) that is directly checked on the lasso guessed in Step 1. We need again Parikh automata to guess a succinct representation of the lasso. The lower bounds for those problems were already known for qualitative reachability games [27]. See the long version [17].

We thus focus on the NCNS problem (Theorem 5.d). We prove below that this problem is EXPTIME-hard, already for two-player environments. The decidability is left open. This decision problem is a real challenge that cannot be solved by known approaches. Indeed, the technique of tree automata, as used in [21] to show the decidability of several ω-regular objectives, is not applicable in the context of quantitative reachability. This is because, while in the scenario of qualitative reachability, the costs are Boolean and can be encoded within the finite state space of a tree automaton, for quantitative reachability, these costs are now integers that are not bounded and vary according to the strategy σ_0 that is being synthesized. Consequently, it is not feasible to directly encode constraints within the states of the automaton in this latter case. Additionally, there is a necessity to enforce constraints related to subtrees, such as comparing (unbounded) costs between two subtrees. Generally, incorporating the capability to enforce subtree constraints in tree automata results in undecidability, with only certain subclasses having a decidable emptiness problem, see e.g. [3]. Therefore, addressing the general case would necessitate either advancements in the field of automata theory or an entirely new methodological approach.

However, we are able to solve the practically relevant case of one-player environments for which we prove that the NCNS problem is PSPACE-hard and in EXPTIME in the long version of this paper [17]. The PSPACE-hardness is given by a classical reduction from the subset-sum game problem [43]. The intuition for the EXPTIME-membership is the following: it consists in finding a play π where $\mathsf{cost}_0(\pi) \leq c$ such that when the only component of the environment deviates from π, either the system inflicts to the deviating play π' a cost for the environment such that $\mathsf{cost}_1(\pi') > \mathsf{cost}_1(\pi')$ meaning that deviating is not profitable, or it ensures a cost for himself such that $\mathsf{cost}_0(\pi') \leq c$. Note that this approach only works for one-player environments.

We are also able to solve the NCNS problem for any number of players in the environment, for the variant where the rational NE responses of the environment aim to ensure costs bounded by a given threshold rather than minimizing these costs (this is also arguably an interesting model of rationality in practice). This is a perspective studied in [39] in the case of NEs for discounted-sum objectives. We show in the long version [17] that this variant is EXPTIME-complete.

▶ **Theorem 14.** *The Non-Cooperative Nash Synthesis problem where the objective of each player $i \in \{1, \dots, t\}$ is a bounded reachability objective $\mathsf{Reach}_{<d_i}(T_i)$ is EXPTIME-complete, and hardness holds even with a one-player environment.*

Reduction for Two-Player Environments. We finally prove that the NCNS problem is EXPTIME-hard, already for a two-player environment (lower bound of Theorem 5.d). The reduction is given from the *countdown game problem*, known to be EXPTIME-complete [31].

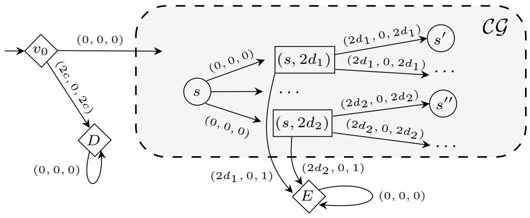

Figure 3 Reduction from the countdown game problem to the NCNS problem (two-player env.).

Given a threshold $c \in \mathbb{N}$, a countdown game $\mathcal{CG}$ is a two-player zero-sum game played on a directed graph (V, E) where $E \subseteq V \times \mathbb{N}^{>0} \times V$. A configuration is a pair $(s, k) \in V \times \mathbb{N}$, initially $(s_0, 0)$ with s_0 an initial vertex, from where player 0 chooses $d \in \mathbb{N}^{>0}$ such that there exists $(s, d, s') \in E$ (we assume that such a d always exists). Player 1 then chooses such an $s' \in V$ to reach the configuration $(s', k + d)$. When reaching a configuration (s, k) with $k \geq c$, the game stops and player 0 wins if and only if $k = c$.[8] Player 0 wins the game $\mathcal{CG}$ if he has a strategy σ_0 from s_0 that allows him to reach some configuration (s, c), whatever the strategy of player 1.

▶ **Theorem 15.** *The Non-Cooperative Nash Synthesis problem with a two-player environment is* EXPTIME*-hard.*

Proof. Given a countdown game $\mathcal{CG}$ and a threshold c, we build a reachability game $\mathcal{G}$ as depicted in Figure 3 with three players, player 0 (owning the circle vertices of $\mathcal{CG}$), player 1 (owning the square vertices of $\mathcal{CG}$), and player 2 (owning the initial vertex v_0 and vertices D, E). The three weight functions are indicated on the edges, with a null weight on all edges for player 1. The initial vertex v_0 has two outgoing edges, one towards vertex D and the other one to the initial vertex s_0 of $\mathcal{CG}$. Inside $\mathcal{CG}$, players 0 and 1 are simulating the countdown game. The target sets are $T_0 = T_2 = \{D, E\}$ and $T_1 = V$. Thus, for any play, player 1 gets a cost of 0 and will never have the incentive to deviate from his strategy. The $\mathcal{CG}$ part of the figure contains a slight modification of the given countdown: players 0 and 1 act as in $\mathcal{CG}$, player 1 can exit it by taking the edge to vertex E, the weights d are multiplied by 2. More precisely, player 0 first selects a transition from a vertex s to some vertex $(s, 2d)$, with $d \in \mathbb{N}^{>0}$, then player 1 responds with a successor s' such that (s, d, s') is an edge in the initial countdown game. At any point $(s, 2d)$, player 1 can exit the $\mathcal{CG}$ by going to E, adding $2d$ to the cost of player 0 and 1 to the cost of player 2, i.e., it gives the cost tuple $(2k + 2d, 0, 2k + 1)$ where $2k$ is the accumulated weight inside $\mathcal{CG}$ before exiting it.

Let us show that a strategy $\sigma_0 \in \Sigma_0$ is a solution to the NCNS problem with the threshold $2c$ if and only if it is winning in the given countdown game and threshold c. We first suppose that σ_0 is a winning strategy for player 0 in the countdown game. We consider this strategy in $\mathcal{G}$ and enumerate all possible plays consistent with σ_0:

- The play $v_0(D)^\omega$ gives the cost $2c$ to player 0, thus satisfying the threshold $2c$,
- No play staying infinitely often in $\mathcal{CG}$ is the outcome of a σ_0-fixed NE, as it gives an infinite cost to player 2 while player 2 could deviate in v_0 to get a cost of $2c < +\infty$,
- Any play π ultimately reaching E has $\mathsf{cost}_0(\pi) = 2k + 2d$ and $\mathsf{cost}_2(\pi) = 2k + 1$, for some $k \in \mathbb{N}$. If $2k + 2d \leq 2c$, then $\mathsf{cost}_0(\pi)$ satisfies the threshold constraint. Otherwise, $2k + 2d > 2c$, but as σ_0 is winning in the initial countdown game, this means that there was a previous configuration where the costs of both players 0 and 2 were exactly $2c$. This means that $\mathsf{cost}_2(\pi) = 2k + 1 \geq 2c + 1$, i.e., π is again not a σ_0-fixed NE outcome.

[8] Classically, the initial configuration is (s_0, c) and the accumulated weight k decreases until being ≤ 0.

Assume now that σ_0 is not winning in the countdown game. Hence, there exists a losing play consistent with σ_0 in this game, that leads to a play π in the grey part of Figure 3 such that in none of its vertices, the accumulated weight is exactly $2c$, i.e., there are two consecutive steps where the accumulated weight is $2k < 2c$ and then $2k + 2d > 2c$. So, player 1 can exit between these two steps to reach E. The resulting play π' has $\mathsf{cost}_0(\pi') = 2k + 2d > 2c$ and $\mathsf{cost}_2(\pi') = 2k + 1 < 2c + 1$, thus $\mathsf{cost}_2(\pi') < 2c$. Consequently, π' is a σ_0-fixed NE outcome but $\mathsf{cost}_0(\pi) > 2c$. It follows that σ_0 is not a solution to the NCNS problem. ◀

6 Conclusion

In this paper, we have determined the exact complexity class for several rational verification and synthesis problems in quantitative reachability games, considering both NE and PO rational behaviors of the environment. However, for the NCNS problem, while we have solved the important one-player environment case, we have left open the multi-player environment case. We believe this latter case poses a significant challenge that may require new advances in automata techniques to be solved.

There are several interesting future works to investigate. (1) We intend to study the FPT complexity of the studied problems. Notice that some of our lower bounds results already hold for one-player environments (see the CNS and UNCNV problems in Section 5). (2) Instead of one reachability objective, player 0 could have several ones and a threshold on these objectives that he wants to see satisfied. (3) The concept of NE could be replaced by SPE or by strong NE (that allows collaborations between the players during deviations). Still, it is important to note that strategies σ_0 that are solutions to the non-cooperative synthesis problems under NE rationality are also solutions under SPE (resp. strong NE) rationality, as SPEs (resp. strong NEs) constitute a subset of NEs.

References

1 Shaull Almagor, Orna Kupferman, and Giuseppe Perelli. Synthesis of controllable nash equilibria in quantitative objective game. In Jérôme Lang, editor, *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pages 35–41. ijcai.org, 2018. doi:10.24963/IJCAI.2018/5.

2 Rajeev Alur, Aldric Degorre, Oded Maler, and Gera Weiss. On Omega-Languages Defined by Mean-Payoff Conditions. In Luca de Alfaro, editor, *Foundations of Software Science and Computational Structures, 12th International Conference, FOSSACS 2009, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2009, York, UK, March 22-29, 2009. Proceedings*, volume 5504 of *Lecture Notes in Computer Science*, pages 333–347. Springer, 2009. doi:10.1007/978-3-642-00596-1_24.

3 Luis Barguñó, Carles Creus, Guillem Godoy, Florent Jacquemard, and Camille Vacher. The Emptiness Problem for Tree Automata with Global Constraints. In *Proceedings of the 25th Annual IEEE Symposium on Logic in Computer Science, LICS 2010, 11-14 July 2010, Edinburgh, United Kingdom*, pages 263–272. IEEE Computer Society, 2010. doi:10.1109/LICS.2010.28.

4 Romain Brenguier and Jean-François Raskin. Pareto Curves of Multidimensional Mean-Payoff Games. In Daniel Kroening and Corina S. Pasareanu, editors, *Computer Aided Verification – 27th International Conference, CAV 2015, San Francisco, CA, USA, July 18-24, 2015, Proceedings, Part II*, volume 9207 of *Lecture Notes in Computer Science*, pages 251–267. Springer, 2015. doi:10.1007/978-3-319-21668-3_15.

5 Léonard Brice, Jean-François Raskin, and Marie van den Bogaard. Subgame-Perfect Equilibria in Mean-Payoff Games. In Serge Haddad and Daniele Varacca, editors, *32nd International Conference on Concurrency Theory, CONCUR 2021, August 24-27, 2021, Virtual Conference*, volume 203 of *LIPIcs*, pages 8:1–8:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPICS.CONCUR.2021.8.

6 Léonard Brice, Jean-François Raskin, and Marie van den Bogaard. The Complexity of SPEs in Mean-Payoff Games. In Mikolaj Bojanczyk, Emanuela Merelli, and David P. Woodruff, editors, *49th International Colloquium on Automata, Languages, and Programming, ICALP 2022, July 4-8, 2022, Paris, France*, volume 229 of *LIPIcs*, pages 116:1–116:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPICS.ICALP.2022.116.

7 Léonard Brice, Jean-François Raskin, and Marie van den Bogaard. Rational Verification for Nash and Subgame-Perfect Equilibria in Graph Games. In Jérôme Leroux, Sylvain Lombardy, and David Peleg, editors, *48th International Symposium on Mathematical Foundations of Computer Science, MFCS 2023, August 28 to September 1, 2023, Bordeaux, France*, volume 272 of *LIPIcs*, pages 26:1–26:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPICS.MFCS.2023.26.

8 Thomas Brihaye, Véronique Bruyère, Aline Goeminne, and Jean-François Raskin. Constrained existence problem for weak subgame perfect equilibria with ω-regular Boolean objectives. *Inf. Comput.*, 278:104594, 2021. doi:10.1016/J.IC.2020.104594.

9 Thomas Brihaye, Véronique Bruyère, Aline Goeminne, Jean-François Raskin, and Marie van den Bogaard. The Complexity of Subgame Perfect Equilibria in Quantitative Reachability Games. In Wan J. Fokkink and Rob van Glabbeek, editors, *30th International Conference on Concurrency Theory, CONCUR 2019, August 27-30, 2019, Amsterdam, the Netherlands*, volume 140 of *LIPIcs*, pages 13:1–13:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPICS.CONCUR.2019.13.

10 Thomas Brihaye, Véronique Bruyère, Aline Goeminne, and Nathan Thomasset. On relevant equilibria in reachability games. *J. Comput. Syst. Sci.*, 119:211–230, 2021. doi:10.1016/J.JCSS.2021.02.009.

11 Thomas Brihaye, Véronique Bruyère, and Gaspard Reghem. Quantitative Reachability Stackelberg-Pareto Synthesis is NEXPTIME-Complete. In Olivier Bournez, Enrico Formenti, and Igor Potapov, editors, *Reachability Problems – 17th International Conference, RP 2023, Nice, France, October 11-13, 2023, Proceedings*, volume 14235 of *Lecture Notes in Computer Science*, pages 70–84. Springer, 2023. doi:10.1007/978-3-031-45286-4_6.

12 Thomas Brihaye, Julie De Pril, and Sven Schewe. Multiplayer Cost Games with Simple Nash Equilibria. In Sergei N. Artëmov and Anil Nerode, editors, *Logical Foundations of Computer Science, International Symposium, LFCS 2013, San Diego, CA, USA, January 6-8, 2013. Proceedings*, volume 7734 of *Lecture Notes in Computer Science*, pages 59–73. Springer, 2013. doi:10.1007/978-3-642-35722-0_5.

13 Thomas Brihaye, Gilles Geeraerts, Axel Haddad, and Benjamin Monmege. To Reach or not to Reach? Efficient Algorithms for Total-Payoff Games. In Luca Aceto and David de Frutos-Escrig, editors, *26th International Conference on Concurrency Theory, CONCUR 2015, Madrid, Spain, September 1.4, 2015*, volume 42 of *LIPIcs*, pages 297–310. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2015. doi:10.4230/LIPICS.CONCUR.2015.297.

14 Thomas Brihaye and Aline Goeminne. Multi-weighted Reachability Games. In Olivier Bournez, Enrico Formenti, and Igor Potapov, editors, *Reachability Problems – 17th International Conference, RP 2023, Nice, France, October 11-13, 2023, Proceedings*, volume 14235 of *Lecture Notes in Computer Science*, pages 85–97, Cham, 2023. Springer Nature Switzerland. doi:10.1007/978-3-031-45286-4_7.

15 Thomas Brihaye, Aline Goeminne, James C. A. Main, and Mickael Randour. Reachability Games and Friends: A Journey Through the Lens of Memory and Complexity (Invited Talk). In Patricia Bouyer and Srikanth Srinivasan, editors, *43rd IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2023, December 18-20, 2023, IIIT Hyderabad, Telangana, India*, volume 284 of *LIPIcs*, pages 1:1–1:26. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPICS.FSTTCS.2023.1.

16 Véronique Bruyère. Synthesis of Equilibria in Infinite-Duration Games on Graphs. *ACM SIGLOG News*, 8(2):4–29, May 2021. doi:10.1145/3467001.3467003.

17 Véronique Bruyère, Christophe Grandmont, and Jean-François Raskin. As soon as possible but rationally. *CoRR*, abs/2403.00399, 2024. doi:10.48550/arXiv.2403.00399.

18 Véronique Bruyère, Jean-François Raskin, and Clément Tamines. Stackelberg-Pareto Synthesis. In Serge Haddad and Daniele Varacca, editors, *32nd International Conference on Concurrency Theory, CONCUR 2021, August 24-27, 2021, Virtual Conference*, volume 203 of *LIPIcs*, pages 27:1–27:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPICS.CONCUR.2021.27.

19 Véronique Bruyère, Jean-François Raskin, and Clément Tamines. Pareto-Rational Verification. In Bartek Klin, Slawomir Lasota, and Anca Muscholl, editors, *33rd International Conference on Concurrency Theory, CONCUR 2022, September 12-16, 2022, Warsaw, Poland*, volume 243 of *LIPIcs*, pages 33:1–33:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.CONCUR.2022.33.

20 Krishnendu Chatterjee, Laurent Doyen, Thomas A. Henzinger, and Jean-François Raskin. Generalized Mean-payoff and Energy Games. In Kamal Lodaya and Meena Mahajan, editors, *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2010, December 15-18, 2010, Chennai, India*, volume 8 of *LIPIcs*, pages 505–516. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2010. doi:10.4230/LIPICS.FSTTCS.2010.505.

21 Rodica Condurache, Emmanuel Filiot, Raffaella Gentilini, and Jean-François Raskin. The Complexity of Rational Synthesis. In Ioannis Chatzigiannakis, Michael Mitzenmacher, Yuval Rabani, and Davide Sangiorgi, editors, *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, volume 55 of *LIPIcs*, pages 121:1–121:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPIcs.ICALP.2016.121.

22 A. Ehrenfeucht and J. Mycielski. Positional strategies for mean payoff games. *International Journal of Game Theory*, 8(2):109–113, June 1979. doi:10.1007/BF01768705.

23 Diego Figueira and Leonid Libkin. Path Logics for Querying Graphs: Combining Expressiveness and Efficiency. In *Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, Kyoto, Japan, July 2015. IEEE. doi:10.1109/LICS.2015.39.

24 Nathanaël Fijalkow and Florian Horn. Les jeux d'accessibilité généralisée. *Tech. Sci. Informatiques*, 32(9-10):931–949, 2013. doi:10.3166/TSI.32.931-949.

25 Emmanuel Filiot, Raffaella Gentilini, and Jean-François Raskin. The Adversarial Stackelberg Value in Quantitative Games. In Artur Czumaj, Anuj Dawar, and Emanuela Merelli, editors, *47th International Colloquium on Automata, Languages, and Programming (ICALP 2020)*, volume 168 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 127:1–127:18, Dagstuhl, Germany, 2020. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ICALP.2020.127.

26 Dana Fisman, Orna Kupferman, and Yoad Lustig. Rational Synthesis. In Javier Esparza and Rupak Majumdar, editors, *Tools and Algorithms for the Construction and Analysis of Systems, 16th International Conference, TACAS 2010, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2010, Paphos, Cyprus, March 20-28, 2010. Proceedings*, volume 6015 of *Lecture Notes in Computer Science*, pages 190–204. Springer, 2010. doi:10.1007/978-3-642-12002-2_16.

27 Christophe Grandmont. Rational Synthesis and Verification in Multiplayer Reachability Games Played on Graphs. Master's thesis, UMONS, June 2023.

28 Julian Gutierrez, Muhammad Najib, Giuseppe Perelli, and Michael J. Wooldridge. Automated temporal equilibrium analysis: Verification and synthesis of multi-player games. *Artif. Intell.*, 287:103353, 2020. doi:10.1016/J.ARTINT.2020.103353.

29 Julian Gutierrez, Muhammad Najib, Giuseppe Perelli, and Michael J. Wooldridge. On the complexity of rational verification. *Ann. Math. Artif. Intell.*, 91(4):409–430, 2023. doi:10.1007/S10472-022-09804-3.

30 David Hyland, Julian Gutierrez, Shankaranarayanan Krishna, and Michael J. Wooldridge. Rational verification with quantitative probabilistic goals. In Mehdi Dastani, Jaime Simão Sichman, Natasha Alechina, and Virginia Dignum, editors, *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2024, Auckland, New Zealand, May 6-10, 2024*, pages 871–879. ACM, 2024. doi:10.5555/3635637.3662941.

31 Marcin Jurdzinski, Francois Laroussinie, and Jeremy Sproston. Model Checking Probabilistic Timed Automata with One or Two Clocks. *Logical Methods in Computer Science*, Volume 4, Issue 3, September 2008. doi:10.2168/LMCS-4(3:12)2008.

32 Felix Klaedtke and Harald Rueß. Monadic Second-Order Logics with Cardinalities. In Jos C. M. Baeten, Jan Karel Lenstra, Joachim Parrow, and Gerhard J. Woeginger, editors, *Automata, Languages and Programming*, pages 681–696, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

33 Orna Kupferman, Giuseppe Perelli, and Moshe Y. Vardi. Synthesis with Rational Environments. In Nils Bulling, editor, *Multi-Agent Systems – 12th European Conference, EUMAS 2014, Prague, Czech Republic, December 18-19, 2014, Revised Selected Papers*, volume 8953 of *Lecture Notes in Computer Science*, pages 219–235. Springer, 2014. doi:10.1007/978-3-319-17130-2_15.

34 Orna Kupferman and Noam Shenwald. The Complexity of LTL Rational Synthesis. In Dana Fisman and Grigore Rosu, editors, *Tools and Algorithms for the Construction and Analysis of Systems*, pages 25–45, Cham, 2022. Springer International Publishing.

35 John F. Nash. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences of the United States of America*, 36:48–49, 1950. doi:10.1073/pnas.36.1.48.

36 Martin J. Osborne. *An introduction to game theory*. Oxford Univ. Press, 2004.

37 Christos H. Papadimitriou and Mihalis Yannakakis. On the Approximability of Trade-offs and Optimal Access of Web Sources. In *41st Annual Symposium on Foundations of Computer Science, FOCS 2000, 12-14 November 2000, Redondo Beach, California, USA*, pages 86–92. IEEE Computer Society, 2000. doi:10.1109/SFCS.2000.892068.

38 Amir Pnueli and Roni Rosner. On the Synthesis of a Reactive Module. In *Conference Record of the Sixteenth Annual ACM Symposium on Principles of Programming Languages, Austin, Texas, USA, January 11-13, 1989*, pages 179–190. ACM Press, 1989. doi:10.1145/75277.75293.

39 Senthil Rajasekaran, Suguman Bansal, and Moshe Y. Vardi. Multi-Agent Systems with Quantitative Satisficing Goals. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI 2023, 19th-25th August 2023, Macao, SAR, China*, pages 280–288. ijcai.org, 2023. doi:10.24963/IJCAI.2023/32.

40 Mickael Randour. Games with multiple objectives. In Nathanaël Fijalkow, editor, *Games on Graphs*. Online, 2023. doi:10.48550/arxiv.2305.10546.

41 Mickael Randour, Jean-François Raskin, and Ocan Sankur. Percentile queries in multi-dimensional Markov decision processes. *Formal Methods Syst. Des.*, 50(2-3):207–248, 2017. doi:10.1007/S10703-016-0262-7.

42 Larry J. Stockmeyer. The polynomial-time hierarchy. *Theoretical Computer Science*, 3(1):1–22, 1976. doi:10.1016/0304-3975(76)90061-X.

43 Stephen Travers. The complexity of membership problems for circuits over sets of integers. *Theoretical Computer Science*, 369(1):211–229, 2006. doi:10.1016/j.tcs.2006.08.017.

A Example of a Nondeterministic Mealy Machine and Product Game

We first provide an example of a nondeterministic Mealy machine and the way it encodes strategies.

▶ **Example 16.** Consider the arena in Figure 4 and the nondeterministic Mealy machine $\mathcal{M}_0$ of player 0 illustrated in Figure 5, formally defined as $\mathcal{M}_0 = (M, m_0, \delta, \tau)$ such that

- $M = \{m_0, m_1\}$,
- $\delta(m_0, v_3) = \{m_0, m_1\}$ and $\delta(m, v) = \{m\}$ for every $(m, v) \neq (m_0, v_3)$,
- $\tau(m_0, v) = \begin{cases} \{v_1, v_3\} & \text{if } v = v_1 \\ \{v_3\} & \text{if } v = v_2 \end{cases}$, and $\tau(m_1, v) = \{v_2\}$ if $v = v_1$ or $v = v_2$.

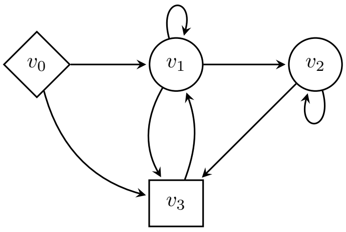

Figure 4 An arena with player 0, player □, and player ◇, with no weight displayed.

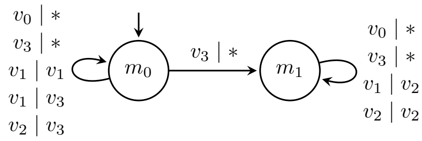

Figure 5 A nondeterministic Mealy machine of player 0. The notation $v \mid v'$ on the transitions (m, m') indicates that $m' \in \delta(m, v)$, and if $v \in V_0$, that $v' \in \tau(m, v)$, otherwise $v' = *$.

The idea is to start and stay in the memory state m_0 and then, once v_3 has been visited, to nondeterministically switch to the memory state m_1, or continue staying in the memory state m_0. The memory state defines which edge player 0 is able to choose from v_1: either a nondeterministic choice between v_1 and v_3 in m_0, or v_2 in m_1.

We now formally define the notion of product arena. Let $\mathcal{A} = (V, E, \mathcal{P}, (V_i)_{i\in\mathcal{P}}, (w_i)_{i\in\mathcal{P}})$ be a weighted arena and $\mathcal{M}_j = (M, m_0, \delta, \tau)$ be a (nondeterministic) Mealy machine for player $j \in \mathcal{P}$. Then, the *product arena* $\mathcal{A} \times \mathcal{M}_j$ is the weighted arena $\mathcal{A} \times \mathcal{M}_j = (V', E', \mathcal{P}, (V'_i)_{i\in\mathcal{P}}, (w'_i)_{i\in\mathcal{P}})$ where

- $V' = (V \times M) \cup (V \times V \times M)$,
- $V'_i = V_i \times M$ for all $i \in \mathcal{P}\setminus\{j\}$, and $V'_j = (V_j \times M) \cup (V \times V \times M)$,
- E' is the set of edges defined as
  - $(v, m) \to (v, v', m)$ if $(v, v') \in E$, and when $v \in V_j$, it must hold that $v' \in \tau(m, v)$,
  - $(v, v', m) \to (v', m')$ if $m' \in \delta(m, v)$,
- For the edges $e' \in E'$ of the form $(v, m) \to (v, v', m)$, $w'_i(e') = w_i((v, v'))$, while for the edges e' of the form $(v, v', m) \to (v', m')$, $w'_i(e') = 0$, for all players $i \in \mathcal{P}$.

Intuitively, in vertices (v, v', m), it is player j who decides how to update the memory state m according to δ.

When $\mathcal{A}$ is initialized with v_0 as initial vertex, then the product arena is also initialized with (v_0, m_0) as initial vertex. Given a reachability game $\mathcal{G} = (\mathcal{A}, (T_i)_{i\in\mathcal{P}})$, we also define the *product game* $\mathcal{G} \times \mathcal{M}_j$ as the reachability game $(\mathcal{A} \times \mathcal{M}_j, (T'_i)_{i\in\mathcal{P}})$ such that $T'_i = T_i \times M$ for all $i \in \mathcal{P}$.

Back to Example 16, the product arena $\mathcal{A}' = \mathcal{A} \times \mathcal{M}_0$ is depicted in Figure 6. We can see that player 0 has several strategies $\sigma_0 \in [\![\mathcal{M}_0]\!]$ whose behavior changes according to the memory state m_0 or m_1.

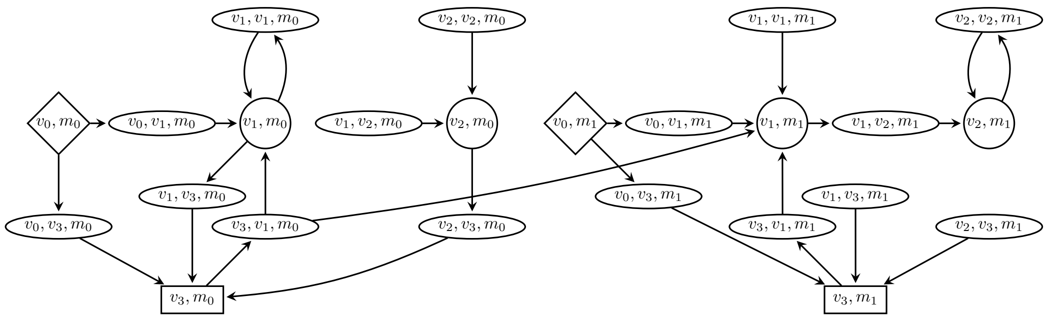

Figure 6 The product arena of the arena in Figure 4 and the Mealy machine in Figure 5.