Synthesis of Resource-Aware Controllers Against Rational Agents

Rodica Condurache
A. I. Cuza University of Iaşi
Iaşi, Romania
rodica.condurache@info.uaic.ro

Catalin Dima
LACL – Université Paris-Est Créteil
Créteil, France
catalin.dima@u-pec.fr

Youssouf Oualhadj
LACL – Université Paris-Est Créteil
Créteil, France
youssouf.oualhadj@u-pec.fr

Nicolas Troquard
KRDB – Free University of Bozen-Bolzano
Bozen-Bolzano, Italy
nicolas.troquard@unibz.it

ABSTRACT

This paper proposes new contributions from the field of formal multiagent systems in the pursued efforts of engineering solutions for the sustainable management of common-pool resources in presence of rational agents.

Non-cooperative rational synthesis is the task of automatically constructing a controller for a reactive system that ensures a given specification against any individually rational behavior of the system's components.

In this paper we consider the case where the controller has to ensure that the system's resources are never depleted. We report complexity results for classical specification such as the one given in linear temporal logic.

KEYWORDS

Rational Synthesis, Non-Cooperative Games, Resources, Logic, Complexity

ACM Reference Format:
Rodica Condurache, Catalin Dima, Youssouf Oualhadj, and Nicolas Troquard. 2023. Synthesis of Resource-Aware Controllers Against Rational Agents. In *Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), London, United Kingdom, May 29 – June 2, 2023*, IFAAMAS, 9 pages.

1 INTRODUCTION

There is a renewed public and institutional interest for a sustainable human development. In particular, the investigation of engineering methods for the governance of common-pool resource systems, or commons [16], has become a pressing need. It is part of Goal 12 of the United Nations, "Ensure sustainable consumption and production patterns". Moreover, the demand for an efficient management of resources has always existed in computing, and also becomes an important aspect for the deployment of autonomous robots way beyond their confined industrial environment.

This paper is a proposal for new generic methods from the field of formal multiagent systems in the pursued efforts of engineering solutions for the sustainable management of the commons in presence of rational agents.

Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), A. Ricci, W. Yeoh, N. Agmon, B. An (eds.), May 29 – June 2, 2023, London, United Kingdom.

Rational synthesis [9, 13, 14] marks a departure from the classical synthesis problem [17] which consists in constructing a reactive system so that the objective of a controller is satisfied in all resulting runs. When the environment is made of rational agents, classical synthesis is considered too pessimistic, because some behaviors of the environment become unreasonable. Instead, *rational synthesis* is the task of constructing automatically a reactive system that satisfies the objective of a controller in all *rational* behaviors of the system's components.

Games and Nash equilibria. The system is composed of several individual acting entities, the *players*, each with their own *objective*. They interact in a turn-based fashion in a *game arena* which consists in a graph, where every node is controlled by one and only one player. In a node that he controls, a player decides the next state of the system. One of the players is distinguished, and is called the *controller*. The objective of the controller corresponds to the specification of the system to be constructed.

A (individually) rational behavior of all the players is an infinite run in the game arena where any player who does not achieve his objective cannot satisfy it by unilaterally changing his own behavior. This corresponds to a *Nash equilibrium* [15].

Rational synthesis. There are two forms of rational synthesis: cooperative and non-cooperative [14]. *Cooperative rational synthesis* can be seen as the synthesis of a Nash equilibrium that satisfies the objective of the controller, and this is akin to the problem of *equilibrium checking* [1]. One must practically convince every player that everyone else is playing their part of the Nash equilibrium. Since no one has an incentive to change their strategy, the Nash equilibrium should be played, and the specification of the system should be satisfied. Non-cooperative rational synthesis is different in that, one does not suggest a strategy to the non-controller players. The problem of *non-cooperative rational synthesis* is the problem of automatically constructing a strategy for the controller such that against any Nash equilibrium which contains the controller's strategy, the specification of the system holds.

Until recently, the rational synthesis focused on qualitative specifications, that is, specifications that either hold or not. But in an effort of designing formal tools that could handle real-life scenarios, the focus started shifting towards the rational synthesis where specifications are also quantitative. One could think of it as an idealization of the notion of resource (energy, money, raw material, …). Among those efforts we refer to [11, 18] where a payoff is assigned to every run of the system, and the goal of synthesis in this case is to design the best possible equilibrium with respect to this payoff. In [6] a different view was considered. A global resource has its amount governed by the transitions taken by the system (and thus by the choices of agents) and the synthesis problem in this context aims at building equilibria that do not deplete the resource in the system. An important distinction in that work that focuses on cooperative rational synthesis is the one of careless and careful players. The notion of resource is specifically the one of a *common-pool resource.*

Common-pool resources. A common-pool resource is a resource that is non-excludable (every agent of the system can consume it) and rivalrous (one agent's consumption of it, limits the opportunity of another another to consume it).

In [6], the problem of cooperative rational synthesis is considered in systems with one *common-pool resource*. By choosing an edge to follow to reach a state, the action of a player may change the amount of the common resource available in the system. The players may be careless or careful. *Careful* players want to achieve their qualitative objective in a run that never depletes the resource. *Careless* players are only interested in their qualitative objective, regardless of the level of the resource.

In this work, we investigate the problem of *non-cooperative* rational synthesis in systems with one common-pool resource, where the specification of the system is a qualitative objective to be realised in a run that never depletes the common-pool resource (the controller is careful in the sense of [6]), and the system's components (apart from the controller) are careless. We do not investigate the case of careful players in our case. Importantly, being careless does not prevent the players to be occasionally acting *as if* they were careful; the opposite is not true.

For convenience, we sometimes call the common-pool resource the *energy* of the system.

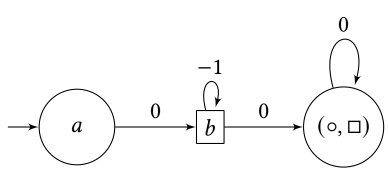

Figure 1: A two-player game without a solution.

Two very simple examples. A very simple game is illustrated in Figure 1. The controller, player 1 controls the circle-states, and player 2 controls the square-state. Both want to reach the state on the right. It is easy to verify that if player 2 is careless, then player 1 does not have a strategy to ensure that for all rational behaviors of player 2, the state on the right is reached with the energy remaining always non-negative. Indeed, if player 2 is careless, then any of his strategies is a rational behavior, including taking the self-loop any finite number of times. Thus, there is no solution to the problem of *non-cooperative feasible* rational synthesis, because player 1 has only one strategy and there are rational behaviors of player 2 that deplete the resource. This is in contrast with the existence of a solution to the *careless cooperative* rational synthesis in the sense of [6]. Namely, player 1 going right, then player 2 going directly to the right, and player 1 looping over the right-most state is a Nash equilibrium, because both players satisfy their objective, and it never depletes the resource.

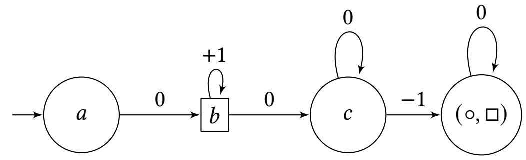

Figure 2: A two-player game with a solution.

Another simple game is illustrated in Figure 2, where a solution to the problem of non-cooperative feasible rational synthesis (with objective to reach the right-most state) exists. Player 1 does not have a winning strategy. Indeed, player 1 wants to reach the right-most state with an energy level staying positive, but this can only happen if player 2 loops over b a few times and eventually goes to c. However, against a rational player 2, one can synthesise a resource-aware controller to satisfy the objective of the system. One solution to the non-cooperative rational synthesis consists in player 1 going right from a to b, and in c:

- going right if the level of the resource is at least 1;
- staying in c otherwise.

In the context of this strategy for player 1, all rational answers by player 2 consist in looping any positive finite number of times over b, and then going to c. All result in a Nash equilibrium, and satisfy the objective of player 1 without depleting the resource. There are no other Nash equilibrium containing this strategy of player 1.

Contribution. We introduce the problem of *non-cooperative feasible rational synthesis* in multi-player turn-based arenas that is the problem of computing a controller that enforces any rational behavior to remain feasible (does not deplete the resource). We show that this problem is 2EXPTIME-complete when the specification of the players are given in LTL, c.f. Theorem 7. In order to establish this result, we adapt the construction from [8] to our case. While this adaptation is rather straightforward, it requires solving a two-player turn-based game with a complex and novel objective; we call *FPP games* the generic class of two-player games with these objectives, c.f. Section 3.3. We study these games in Section 4. The key steps are represented on Figure 3. We show that the problem of deciding the winner in a FPP game lies in NP ∩ co-NP. This result is obtained through an encoding into the so-called *energy parity games* [4]. Finally, we enrich the specification of controller by adding an LTL specification. In this case, a solution must ensure that any rational behavior of the players is not only feasible but that satisfies an LTL specification, we show that this problem reduces to the non-cooperative feasible rational synthesis and that it remains 2EXPTIME-complete c.f. Section 5.2.

Outline. We present the settings in Section 2. We define the class of multi-player games and the notion of fixed-Nash equilibria within them. We introduce the various kinds of objectives that we are considering in this paper. We also define the problem of non-cooperative feasible rational synthesis. In Section 3, we present the reduction to the problem of finding a winning strategy in a

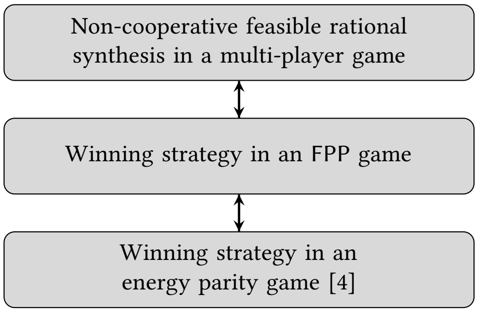

Figure 3: Chain of reductions used in solving the non-cooperative feasible rational synthesis problem.

two-player turn-based zero-sum game that we call the negotiation game. In Section 4 we relate solving such a game to a new class of turn-based games called FPP game. A solution is presented in Proposition 5. Finally in Section 5 we wrap up our results by establishing complexity bounds for the non-cooperative feasible rational synthesis problem. We also show that the complexity of the problem is unchanged whether the controller has a qualitative specification together with a feasibility goal or not. We finally conclude in Section 6.

2 GAMES ON FINITE GRAPHS

For any set Q we denote by Q^* the set of finite sequences of elements in Q and Q^ω the set of infinite sequences of elements of Q. Let $w \in Q^* \cup Q^\omega$, and $i \geq 1$, we denote by $w[i]$ the i-th element in w; we denote by $w[..i]$ the prefix of w of size i and $w[i..]$ the suffix that starts at the i-th letter. For an element $q \in Q^*$, $\mathsf{last}(q)$ is the last element in the sequence q.

2.1 Arenas, Strategies, and Profiles

Multi-player arenas. A *multi-player arena* is a tuple $\mathcal{G} = \langle \mathsf{S}, (\mathsf{S}_1 \uplus \ldots \uplus \mathsf{S}_n), s_{\mathsf{ini}}, \mathsf{P}, \mathsf{E} \rangle$, where S is a finite set of states, $(\mathsf{S}_1 \uplus \ldots \uplus \mathsf{S}_n)$ is a partition of S, s_{ini} is an initial state, $\mathsf{P} = \{1, \ldots, n\}$ is the set of players, and E is in an edge relation in $\mathsf{S} \times \mathsf{S}$. For every edge $e = (s, t)$, $\mathsf{Src}(e)$ is s and $\mathsf{Trgt}(e)$ is t.

Plays and strategies. For an arena $\mathcal{G}$, we denote by $\mathsf{Play}(\mathcal{G})$ the set of elements $s_{\mathsf{ini}} s_1 s_2 \ldots$ in S^ω such that for all $i \geq 0$, (s_i, s_{i+1}) is in E. The set $\mathsf{Hist}(\mathcal{G})$ is the set of finite and proper prefixes of elements in $\mathsf{Play}(\mathcal{G})$. Moreover, the set of histories for player i denoted $\mathsf{Hist}_i(\mathcal{G})$ is the set of elements in $\mathsf{Hist}(\mathcal{G})$ whose last element is in S_i i.e., $\mathsf{Hist}_i(\mathcal{G}) = \{h \in \mathsf{Hist}(\mathcal{G}) \mid \mathsf{last}(h) \in \mathsf{S}_i\}$. A *strategy* for player i is a function $\sigma_i \colon \mathsf{Hist}_i(\mathcal{G}) \to \mathsf{S}$ mapping a history h, whose last element s, to a state s' such that $(s, s') \in \mathsf{E}$. For a strategy σ_i for player i, we define the set $\mathsf{out}(\sigma_i, s)$ as the set of plays that are *compatible* with σ_i from state s i.e.,

$$\{\pi \in \mathsf{S}^\omega \mid \pi[0] = s, \text{ and } \forall j \geq 0,\ s.t.\ \pi[j] \in \mathsf{S}_i \implies \sigma_i(\pi[..j]) = \pi[j+1]\} \ .$$

For any play π in $\mathsf{out}(\sigma_i, s)$ we say that π is compatible with the strategy σ_i. Often for the sake of convenience, we will write $\mathsf{out}(\sigma_i)$ to intend $\mathsf{out}(\sigma_i, s_{\mathsf{ini}})$. We also say that a history h is compatible with a strategy σ if h is a prefix of some play π in $\mathsf{out}(\sigma)$.

Profile of strategies. Once a strategy σ_i for each player i is chosen, we obtain a strategy profile $\overline{\sigma} = \langle \sigma_1, \ldots, \sigma_n \rangle$. $\overline{\sigma}_{\text{-}i}$ is the corresponding partial profile without the strategy for player i. For a strategy σ'_i for a player i, we write $\langle \overline{\sigma}_{\text{-}i}, \sigma'_i \rangle$ the profile $\langle \sigma_1, \ldots, \sigma'_i, \ldots, \sigma_n \rangle$. We denote by $\mathsf{out}(\overline{\sigma})$ the unique outcome of the strategy profile $\overline{\sigma}$. That is the play π which is compatible with all the strategies in $\overline{\sigma}$.

2.2 Objectives and Payoffs

We describe the specifications by means of objectives. Regarding the controller, the specification is described by a *feasibility* objective. The other players will be assigned a temporal specification induced by a linear logic formula.

Broadly speaking, an objective Obj is a subset of $\mathsf{Play}(\mathcal{G})$. We write Obj_i to specify that it is the objective of player i. We define the payoff $\mathsf{Payoff}_i(\overline{\sigma})$ of player i w.r.t. the profile $\overline{\sigma}$ as follows: $\mathsf{Payoff}_i(\overline{\sigma}) = 1$ if $\mathsf{out}(\overline{\sigma})$ is in Obj_i and 0 otherwise. In the case where the arena consists of only two players, we can define a zero-sum game, in this case, the objectives of the players are opposed one to another, i.e.:

$$\forall i \in \{1, 2\},\ \mathsf{Obj}_{3-i} = \mathsf{Play}(\mathcal{G}) \setminus \mathsf{Obj}_i \ .$$

Once an arena $\mathcal{G}$ is equipped with an objective Obj_i for each player i, we will often call *game* the tuple $\langle \mathcal{G}, \mathsf{Obj}_1, \ldots, \mathsf{Obj}_n \rangle$. When the objective is clear from the context we will simply write $\mathcal{G}$.

In the context of a zero-sum game, we can define the notion of a *winning* strategy for player i from a state s, i.e., a strategy σ_i s.t. $\mathsf{out}(\sigma_i)$ is a subset of Obj_i. We also define the set of winning states w.r.t. to Obj_1 as follows:

$$\mathsf{Win}(\mathsf{Obj}_i) = \{s \in \mathsf{S} \mid \exists \sigma_1 \text{ s.t. } \mathsf{out}(\sigma_1, s) \subseteq \mathsf{Obj}_i\} \ .$$

Given a a multi-player arena $\mathcal{G} = \langle \mathsf{S}, (\mathsf{S}_1 \uplus \ldots \uplus \mathsf{S}_n), s_{\mathsf{ini}}, \mathsf{P}, \mathsf{E} \rangle$, we write $\mathcal{G}^{\text{-}i}$ for the zero-sum game where player 1 is i, and player 2 is the coalition of the rest of the players seen as one entity. Formally $\mathsf{S}_1 = \mathsf{S}_i$, $\mathsf{S}_2 = \bigcup_{j \neq i} \mathsf{S}_j$, $\mathsf{Obj}_1 = \mathsf{Obj}_i$, and $\mathsf{Obj}_2 = \mathsf{Play}(\mathcal{G}) \setminus \mathsf{Obj}_1$.

In order to describe a feasibility objective we equip the arena with a *cost* function.

Feasibility objectives. Let $\mathsf{cost} \colon \mathsf{E} \to \mathbb{Z}$ be a function. To lighten the notation, we write $\mathsf{cost}(s, t)$ instead of $\mathsf{cost}((s, t))$. Let $h = s_{\mathsf{ini}} \ldots s_n$ be a history in $\mathsf{Hist}(\mathcal{G})$; we abusively write $\mathsf{cost}(h)$ to mean the extension of cost to histories that is: $\mathsf{cost}(h) = \mathsf{cost}(s_{\mathsf{ini}}, s_1) + \sum_{i=1}^{n-1} \mathsf{cost}(s_i, s_{i+1})$.

The set of feasible plays in a game $\mathcal{G}$ equipped with a cost function cost is given by the set Feas described as follows:

$$\mathsf{Feas} = \{\pi \in \mathsf{Play}(\mathcal{G}) \mid \forall i \geq 1,\ \mathsf{cost}(\pi[..i]) \geq 0\} \ .$$

We denote by W the largest absolute value that appears in cost, i.e. $\mathsf{W} = \max\{|c| \in \mathbb{Z} \mid \exists e \in \mathsf{E},\ \mathsf{cost}(e) = c\}$. Throughout the paper, values of cost are encoded in binary, thus W is exponential in its encoding which is $\log(\mathsf{W})$.

Linear Temporal Logic Objectives. Linear Temporal Logic *(LTL)* for short is the set of formulas ϕ defined using the following grammar over a set of atomic proposition AP:

$$\phi ::= \alpha \mid \neg\phi \mid \phi \vee \phi \mid \mathsf{X}\, \phi \mid \phi\, \mathsf{U}\, \phi \,,$$

where α is in AP. In our case we simply assume that AP is formed by the of states of the arena that is S.

Let $\mathcal{G}$ be an arena and lbl be labelling function that maps states with subset of AP. LTL formulas are evaluated over plays of $\mathcal{G}$, as follows:

$$\pi \models \alpha \text{ iff } \alpha \in \mathsf{lbl}(\pi[0])\ ,\ \pi \models \neg\phi \text{ iff } \pi \not\models \phi\ ,$$
$$\pi \models \mathsf{X}\,\phi \text{ iff } \pi[1..] \models \phi\ ,\ \pi \models \phi \vee \psi \text{ iff } \rho \models \phi \text{ or } \pi \models \psi\ ,$$
$$\pi \models \phi\ \mathsf{U}\ \psi \text{ iff } \exists i \geq 0,\ \pi[i..] \models \psi \text{ and } \forall j < i,\ \pi[j..] \models \phi\ ,$$

where $\rho \in \mathsf{Play}(\mathcal{G})$, $\alpha \in \mathsf{AP}$, $\phi \in$ LTL, and $\psi \in$ LTL.

In the sequel, we will use the following macros:

$$\mathsf{F}\phi = \top\ \mathsf{U}\ \phi\ , \qquad\qquad \mathsf{G}\phi = \neg\mathsf{F}\neg\phi\ .$$

The notion $\mathsf{F}\phi$ denotes the set of plays where the formula ϕ holds in some position while the notation $\mathsf{G}\phi$ denotes the set of plays where the formula ϕ holds in every position. Using these notation we define the macro $\mathsf{FG}\phi$, i.e., the set of plays along which ϕ holds at infinitely many positions.

Let Obj be an LTL objective given by a formula ϕ in an arena $\mathcal{G}$ equipped with a labelling function lbl. The objective ϕ induces the following set

$$\mathsf{Obj} = \{\pi \in \mathsf{Play}(\mathcal{G}) \mid \pi \models \phi\}\ .$$

Parity Objectives. In order to study the synthesis for LTL a useful formalism is the one of *parity games*, therefore, we define a parity objective as follows. Let π in $\mathsf{Play}(\mathcal{G})$, we denote by $\mathsf{Inf}(\pi)$ the set of states occurring infinitely often along π. Let C be a finite subset of $\mathbb{N}$, and let $\mathsf{prty}\colon \mathsf{S} \to C$ be a priority function. The parity objective for a game $\mathcal{G}$ equipped with the priority function prty is given by the set Parity defined as follows

$$\mathsf{Parity} = \left\{\pi \in \mathsf{Play}(\mathcal{G}) \mid \min\{\mathsf{prty}(s) \mid s \in \mathsf{Inf}(\pi)\} \text{ is even}\right\}\ .$$

2.3 Non-Cooperative Rational Synthesis

Before stating the problem we are interested in, we formalise the notion of *rational* behavior. We will use the notion of *fixed Nash equilibrium.* A strategy profile $\overline{\sigma}$ is a fixed Nash equilibrium (f-NE) if for any strategy σ_i' for any player i in $\mathsf{P} \setminus \{1\}$ we have: $\mathsf{Payoff}_i(\overline{\sigma}) \geq \mathsf{Payoff}_i(\langle\overline{\sigma}_{-i}, \sigma_i'\rangle)$. We write $\mathsf{f\text{-}NE}(\mathcal{G})$ for the set of all the profiles that are fixed Nash equilibria in $\mathcal{G}$.

Non-Cooperative Feasible Rational Synthesis. Given an arena $\mathcal{G} = \langle \mathsf{S}, (\mathsf{S}_1 \uplus \ldots \uplus \mathsf{S}_n), s_{\mathsf{ini}}, \mathsf{P}, \mathsf{E}\rangle$ and objectives $\mathsf{Obj}_1, \ldots, \mathsf{Obj}_n$ the *co-operative rational synthesis* problem is to decide whether there exists a strategy σ_1 such that for all strategies $\sigma_2, \ldots, \sigma_n$ the profile $\overline{\sigma} = \langle\sigma_1, \ldots, \sigma_n\rangle$ satisfies the following:

$$\mathsf{out}(\overline{\sigma}) \in \mathsf{f\text{-}NE}(\mathcal{G}) \implies \mathsf{out}(\overline{\sigma}) \in \mathsf{Obj}_1\ . \tag{1}$$

In this paper we are interested in the setting where the controller is resource-aware, therefore we equip the arena $\mathcal{G}$ with a cost function cost. In this case the problem is to decide whether there exists a strategy σ_1 such that for all strategies $\sigma_2, \ldots, \sigma_n$ the profile $\overline{\sigma} = \langle\sigma_1, \ldots, \sigma_n\rangle$ satisfies the following:

$$\mathsf{out}(\overline{\sigma}) \in \mathsf{f\text{-}NE}(\mathcal{G}) \implies \mathsf{out}(\overline{\sigma}) \in \mathsf{Feas}\ . \tag{2}$$

Indeed since player 1 a.k.a. the controller is concerned with the resource, he wants to guarantee the feasibility of any rational behavior of the players. The objectives $\mathsf{Obj}_2, \ldots, \mathsf{Obj}_n$ are induced by LTL formulas. We shall call the problem consisting in designing a controller satisfying (2) the *non-cooperative feasible rational synthesis.* We shall also call a controller that is a solution a *resource-aware controller.*

Definition 1. *Given a multi-player arena* $\mathcal{G} = \langle \mathsf{S}, (\mathsf{S}_1 \uplus \ldots \uplus \mathsf{S}_n), s_{\mathsf{ini}}, \mathsf{P}, \mathsf{E}\rangle$ *and objectives* $\mathsf{Obj}_2, \ldots, \mathsf{Obj}_n$*, the problem of* non-cooperative feasible rational synthesis *asks whether the controller has a strategy* σ_1 *such that for all strategies* $\sigma_2, \ldots, \sigma_n$ *of the other players the profile* $\overline{\sigma} = \langle\sigma_1, \ldots, \sigma_n\rangle$ *satisfies Equation 2.*

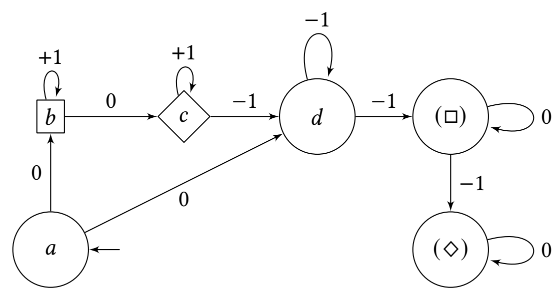

Figure 4: A three-player game. Player 1 has a vacuous qualitative objective and thus only cares about not depleting the resource.

Example 1. *In the game illustrated on Figure 4, there is a solution to the problem of* non-cooperative feasible rational synthesis. *Suppose that player 1 (controls circle states) is only interested in maintaining the energy of the system non-negative. Player 2 and player 3 control the square and diamond states, respectively, and want to reach the state labeled □ and ◇, respectively, regardless of the energy level (i.e., their objective are induced by the LTL formulas* $\mathsf{F}\Box$ *and* $\mathsf{F}\Diamond$*).*

Player 1 does not have a winning strategy to ensure that the energy remains above zero. Indeed, from a*, he must go to* b *where player 2 could go straight to* c *and player 3 could go straight to* d*, bringing the level of energy below zero. The alternative move from* a *to go to* d *necessarily brings the energy below zero at the following step. A strategy* σ_1 *for player 1 which is a solution to the problem of non-cooperative feasible rational synthesis can be informally described as follows:*

- *in* a *go to* b*;*
- *in* d *go to the state labeled □ if the energy is at least 1, and loop otherwise;*
- *in the state labeled □ go to the state labeled ◇ if the energy is at least 1, and loop otherwise;*
- *in the state labeled ◇, loop.*

When player 1 plays σ_1*, one possible Nash Equilibrium is when both player 1 and player 2 loop over* c *and* d*, respectively, for ever. This is a Nash equilibrium, because even though both player 2 and player 3 lose (they do not reach either □ and ◇), they do not have a unilateral deviation to do so. Still, the energy remains above zero, and player 1 satisfies his objective. Another Nash Equilibrium is when player 2 and player 3 loop over* c *and* d *a finite number of times and for a total of at least 3 times. In these cases, the game reaches the state labelled ◇, where all player satisfy their objectives, including player 1 who satisfy his energy objective.*

Before we move on to the resolution of our new synthesis problem, we emphasize that player 1 does not have a qualitative objective. We will later introduce the problem of non-cooperative feasible rational synthesis *with rich specifications* where the system's specification also includes a qualitative objective, and we will show that we can solve it in a uniform manner.

3 COMPUTING A RESOURCE-AWARE CONTROLLER

In this section, we adapt a proof technique initially proposed in [3] to synthesize Nash equilibria in concurrent games with ω-regular objectives. It was later successively adapted to solve the problem of non-cooperative rational synthesis in turn-based games [7] and in concurrent games [8]. We will follow more closely the presentation of the latter.

The technique consists in constructing a turn-based two-player game (we call it the "negotiation game") which is an abstraction of the original multi-player game, in such a way that there is a winning strategy in the abstraction if and only if there a solution to the non-cooperative rational synthesis in the original game. This is done in Section 3.1.

The construction results in a turn-based two-player game with objectives (a Boolean combination of two parity objectives and an energy objective) for which no solution exists. We work out a solution of this class of games in Section 3.3.

3.1 The Negotiation Arena

Given a game arena $\mathcal{G} = \langle \mathsf{S}, (\mathsf{S}_1 \uplus \ldots \uplus \mathsf{S}_n), s_{\mathsf{ini}}, \mathsf{P}, \mathsf{E}\rangle$ we construct a *turn-based 2-player zero-sum* game arena $\mathcal{H} = \langle \widehat{\mathsf{S}}, (\widehat{\mathsf{S}}_\mathsf{C} \uplus \widehat{\mathsf{S}}_\mathsf{S}), \widehat{s}_{\mathsf{ini}}, \widehat{\mathsf{E}}\rangle$ in which Constructor and Spoiler play, and where:

- $\widehat{\mathsf{S}}_\mathsf{C}$ is the set of states controlled by Constructor,
- $\widehat{\mathsf{S}}_\mathsf{S}$ is the set of states controlled by Spoiler,
- $\widehat{s}_{\mathsf{ini}}$ is the initial state,
- $\widehat{\mathsf{E}}$ is the transition table defined over $\widehat{\mathsf{S}} \times \widehat{\mathsf{S}}$.

Let W and D be two subsets of P. The set $\widehat{\mathsf{S}}$ is:

$$\begin{aligned}\widehat{\mathsf{S}} = &\left(\mathsf{S} \times (2^{\mathsf{P}})^2\right)\\ &\cup \left(\mathsf{S} \times (2^{\mathsf{P}})^2 \times (\mathsf{S} \cup \{-\})\right)\\ &\cup \left(\mathsf{S} \times (2^{\mathsf{P}})^2 \times (\mathsf{S} \cup \{-\})\right) \times \mathsf{S} \ .\end{aligned}$$

The set $\widehat{\mathsf{S}}_\mathsf{C}$ is:

$$\begin{aligned}\widehat{\mathsf{S}}_\mathsf{C} = &\{(s, W, D) \mid (s \in \mathsf{S}_1)\}\\ &\cup \{(s, W, D) \mid (s \in \mathsf{S}_{i\neq 1} \wedge i \in W)\}\\ &\cup \{(s, W, D, t) \mid (s \in \mathsf{S}_{i\neq 1}) \wedge (i \notin W \cup D)\} \ .\end{aligned}$$

The set $\widehat{\mathsf{S}}_\mathsf{S}$ is $\widehat{\mathsf{S}} \setminus \widehat{\mathsf{S}}_\mathsf{C}$.
$\widehat{\mathsf{E}}$ contains the following set of transitions:

$$\begin{aligned}&(s, W, D) \mapsto (t, W, D) \text{ if } (s \in \mathsf{S}_1) \wedge ((s, t) \in \mathsf{E}) \ ,\\ &(s, W, D) \mapsto (t, W, D) \text{ if } (s \in \mathsf{S}_i) \wedge (i \in D) \wedge ((s, t) \in \mathsf{E}) \ ,\\ &(s, W, D) \mapsto (s, W, D, t) \text{ if } (s \in \mathsf{S}_{i\neq 1}) \wedge (i \in W) \wedge ((s, t) \in \mathsf{E}) \ ,\\ &(s, W, D) \mapsto (s, W, D, t) \text{ if}\\ &\qquad (s \in \mathsf{S}_i) \wedge \big((i \notin W \cup D) \wedge ((s, t) \in \mathsf{E})\big) \ ,\\ &(s, W, D, t) \mapsto (s, W, D, t, r) \text{ if } (s, r) \in \mathsf{E} \ ,\\ &(s, W, D, t) \mapsto (s, W, D, t, -) \ ,\\ &(s, W, D, t, -) \mapsto (t, W, D) \ ,\\ &(s, W, D, t, r) \mapsto (q, W', D') \ ,\end{aligned}$$

where

$$\begin{aligned}W' &= W \cup \{i\} \text{ if } (q = r) \wedge (s \in \mathsf{S}_i) \ ,\\ D' &= D \cup \{i\} \text{ if } (q = t) \wedge (s \in \mathsf{S}_i) \ .\end{aligned}$$

$\widehat{\mathsf{cost}}$ is defined as follows:

$$\begin{aligned}\big((s, W, D), (t, W, D)\big) &\mapsto \mathsf{cost}(s, t) \ ,\\ \big((s, W, D, t, r), (r, W', D')\big) &\mapsto \mathsf{cost}(s, r) \ ,\\ \big((s, W, D, t, -), (t, W, D)\big) &\mapsto \mathsf{cost}(s, t) \ ,\\ &0 \text{ in all the other cases.}\end{aligned}$$

3.2 Building a Solution

We equip the game arena $\mathcal{G} = \langle \mathsf{S}, (\mathsf{S}_1 \uplus \ldots \uplus \mathsf{S}_n), s_{\mathsf{ini}}, \mathsf{P}, \mathsf{E}\rangle$ with objectives to obtain the multi-player game $\langle \mathcal{G}, \mathsf{Obj}_1, \ldots, \mathsf{Obj}_n\rangle$.

In the negotiation arena, we call the set of states in $\mathsf{S} \times \left(2^{\mathsf{P}}\right)^2 \times (\mathsf{S} \cup \{-\})$ *negotiation* states. The states in $\mathsf{S} \times \left(2^{\mathsf{P}}\right)^2$ are *decision* states. A play in the negotiation arena is a sequence of states. For the ease of notation we will consider the projection of this sequence over the decision states. We denote this set of sequences by $\widehat{\mathsf{S}}^\omega \restriction_{dec}$. We equip $\widehat{\mathsf{S}}$ with the canonical projection proj_i that is the projection over the i-th component. In particular, for every $(s, W, D) \in \widehat{\mathsf{S}}$, we have $\mathsf{proj}_1((s, W, D)) = s$, $\mathsf{proj}_2((s, W, D)) = W$, and $\mathsf{proj}_3((s, W, D)) = W$. We also extend proj_i over $\widehat{\mathsf{S}}^+$ and $\widehat{\mathsf{S}}^\omega$ as expected. The set $\overrightarrow{\mathsf{proj}_2}(\pi \restriction_{\widehat{\mathsf{S}}^1_\mathsf{C}})$ (resp. $\overrightarrow{\mathsf{proj}_3}(\pi \restriction_{\widehat{\mathsf{S}}^1_\mathsf{C}})$) is the set of agents in the limit of W's (resp. D's).

We define the following sets:

$$\begin{aligned}\widehat{\mathsf{Obj}}_\mathsf{D} = \{\pi \in \widehat{\mathsf{S}}^\omega \mid \exists p \in \overrightarrow{\mathsf{proj}_2}(\pi \restriction_{dec}),&\\ \mathsf{proj}_1(\pi \restriction_{dec}) \notin \mathsf{Obj}_p\}& \ , \qquad (3)\\ \widehat{\mathsf{Obj}}_\mathsf{W} = \{\pi \in \widehat{\mathsf{S}}^\omega \mid \forall p \in \overrightarrow{\mathsf{proj}_3}(\pi \restriction_{dec}),&\\ \mathsf{proj}_1(\pi \restriction_{dec}) \in \mathsf{Obj}_p\}& \ , \qquad (4)\\ \widehat{\mathsf{Feas}} = \{\pi \in \widehat{\mathsf{S}}^\omega \mid \widehat{\mathsf{cost}}(\pi) \geq 0\}& \ . \qquad (5)\end{aligned}$$

Finally, we obtain the negotiation game by equipping the negotiation arena with the following winning condition:

$$\big((\widehat{\mathsf{Feas}} \cup \widehat{\mathsf{Obj}}_\mathsf{D}) \cap \widehat{\mathsf{Obj}}_\mathsf{W}\big) \ . \qquad (6)$$

We briefly explain how it relates to analogous approaches in the literature. In [8], a very similar construction is presented in the case of concurrent arenas but with temporal objectives. Here we have adapted this construction to the simpler setting of turn-based games. It may appear that the case of turn-based games was already handled in [7]. However, in the latter work, the approach consists in building a tree automaton. In our case, where the objective of player 1 is quantitative, this would lead to a new class of weighted tree automata for which we would need to solve the emptyness problem. In turn, solving the emptiness of this class of automata would require to solve a new class of turn-based games where the objective is $(\widehat{\mathsf{Feas}} \cup \widehat{\mathsf{Obj}}_\mathsf{D}) \cap \widehat{\mathsf{Obj}}_\mathsf{W}$. Instead, we directly build this turn-based game and extract a solution, by adapting the construction from [8]. We solve this new class of zero-sum two-player games in Section 4.

The intuition behind the winning condition is the following: Constructor aims at building a solution therefore, if he designs a strategy that ensures Feas then clearly this strategy describes a solution. In the case where Feas is not achieved, Constructor has to prove that the players are not behaving rationally. This is where $\widehat{\mathsf{Obj}}_\mathsf{D}$ and $\widehat{\mathsf{Obj}}_\mathsf{W}$ come into play. Indeed $\widehat{\mathsf{Obj}}_\mathsf{D}$ ensures that Constructor has detected the possible deviators, while $\widehat{\mathsf{Obj}}_\mathsf{W}$ prevents him from detecting spurious deviations, i.e., prevents Constructor from falsely suspecting a deviation from a player.

Since the players from 2 to n have qualitative objectives, the sets $\widehat{\mathsf{Obj}}_\mathsf{D}$ and $\widehat{\mathsf{Obj}}_\mathsf{W}$ are exactly those described in [8], and thus fill the same role in the construction and the proof. From the results of [8], it follows that they witness plays in $\mathcal{G}$ that are not outcomes of a rational behavior of players $2 \dots n$.

Finally, the proof from [8] involves two mappings. One that maps the plays in $\mathcal{G}$ into plays in $\mathcal{H}$ and another one mapping plays from $\mathcal{H}$ into plays in $\mathcal{G}$. A crucial property of these two mappings is that applied to our setting, the costs along the plays are preserved, and thus the feasibility of a play in $\mathcal{G}$ witnesses the feasibility of a play in $\mathcal{H}$ and vice versa.

These elements put together entail the following proposition.

Proposition 2. *There exists a solution to the problem of non-cooperative rational synthesis in the multi-player game $\mathcal{G}$ if and only if there exists a winning strategy for* Constructor *in $\mathcal{H}$.*

3.3 Solving the Negotiation Game

In order to solve the negotiation game we need to design an algorithm for a two-player game where the winning condition is given by Equation (6). We proceed as follows; we use results from [4] on solving energy parity games, by first encoding the winning condition with an LTL formula. Then we transform this formula into a parity condition. At this stage one has to make sure that the formula does not blow up, actually the crux is to encode the winning condition of Equation (6) by a formula of size polynomial in $\mathcal{G}$. We recall that $\mathcal{G}$ is equipped with a labelling function $\mathsf{lbl}\colon \mathsf{S} \to 2^{\mathsf{AP}}$ where AP is a set of atomic propositions. We equip the Negotiation Game with a labeling function $\widehat{\mathsf{lbl}}$ which maps elements from $\widehat{\mathsf{S}}$ to subsets of $\widehat{\mathsf{AP}}$ a fresh set of atomic propositions. This new set of atomic propositions is :

$$\widehat{\mathsf{AP}} = \mathsf{AP} \cup \{p_D, p_W \mid p \in \mathsf{P}\} \ ,$$

and the mapping $\widehat{\mathsf{lbl}}$ is defined for each state $s \in \widehat{\mathsf{S}}$ as follows:

$$\widehat{\mathsf{lbl}}(s) = \{p_D \mid p \in \mathsf{proj}_2(s)\} \cup \{p_W \mid p \in \mathsf{proj}_2(s)\} \cup \mathsf{lbl}(\mathsf{proj}_1(s)) \ .$$

Then the set of plays defined in Equations (3), resp. (4), can be characterized by the following LTL formulas:

$$\widehat{\mathsf{Obj}}_\mathsf{D} \equiv \bigvee_{p \in \mathsf{P}} \left(\mathsf{FG} p_D \to \neg \mathsf{Obj}_{p_D}\right) \ , \tag{7}$$

$$\widehat{\mathsf{Obj}}_\mathsf{W} \equiv \bigvee_{p \in \mathsf{P}} \left(\mathsf{FG} p_W \leftrightarrow \neg \mathsf{Obj}_{p_W}\right) \ . \tag{8}$$

Note that the size of both these formulas is polynomial in the size of the original arena $\mathcal{G}$.

Then we apply classical results from [10] to obtain two deterministic parity automata $\mathcal{A}_1$ and $\mathcal{A}_2$, recognizing the sets induced by $\widehat{\mathsf{Obj}}_\mathsf{D}$, resp $\widehat{\mathsf{Obj}}_\mathsf{W}$. The size of both these automata is doubly exponential in the size of the formulas for $\widehat{\mathsf{Obj}}_\mathsf{D}$ resp. $\widehat{\mathsf{Obj}}_\mathsf{W}$ that is, $O\left(2^{2^{|\mathcal{G}|}}\right)$, where $|\mathcal{G}|$ is the size of the description of $\mathcal{G}$. Finally, by a synchronous composition of both automata with the arena $\mathcal{G}$ (synchronized on the atomic propositions $\widehat{\mathsf{AP}}$), we get a new two-player game whose size is $O\left(2^{2^{|\mathcal{G}|}}\right)$ in which the winning condition for the Constructor can be expressed as:

$$(\mathsf{Feas} \cup \mathsf{Parity}_1) \cap \mathsf{Parity}_2 \ . \tag{9}$$

Remark 3. *We highlight that in the above game, the sets* Parity_1 *and* Parity_2 *are induced by priority functions* prty_1 *and* prty_2 *from the automata $\mathcal{A}_1$ and $\mathcal{A}_2$. A crucial property of* prty_1 *and* prty_2 *is that their size is polynomial in the size of formulas of Equations* (7) *and* (8).

We shall call a two-player arena where the winning condition is given by Equation 9 an *FPP game*. We present how to compute winning strategies in such games in Section 4.

4 FPP GAMES

In this part of the paper we design an algorithm for computing a winning strategy for player 1 in an FPP game when it exists. Formally we are given a two-player arena $\mathcal{G} = \langle \mathsf{S}, (\mathsf{S}_1 \uplus \mathsf{S}_2), s_{\mathsf{ini}}, \mathsf{E} \rangle$ equipped with two priority functions prty_1 and prty_2 and a cost function cost. These functions induce three objectives respectively Parity_1, Parity_2, and Feas. As explained earlier, we aim at solving games where the objective is given by the set described in Equation (9).

Before designing a solution to these games, we establish a technical lemma. This lemma is instrumental. It describes a special property of winning strategies in FPP games. Later, we use this property to reduce any FPP game to a game where the winning condition is given by the set $\mathsf{Feas} \cup \mathsf{Parity}$ for some cost function and some priority function. The solution follows from the fact that winning strategy from the initial state in the new game for the objective $\mathsf{Feas} \cap \mathsf{Parity}$ witnesses the existence of a winning strategy in the original game. $\mathsf{Feas} \cap \mathsf{Parity}$ were originally introduced in [4], under the name *energy parity* games, here they are called differently in order to remain consistent with our notation. We also highlight the fact that problem solved in [4] is the initial credit problem that it, computing the least possible energy level in the initial state such that a winning strategy exists for player 1. Note that in this paper the initial credit is always 0, but this does not change the complexity results obtained in [4].

Lemma 4. *Let $\mathcal{G}$ be a two-player arena, σ_1 be a strategy for player 1, and let $X = \mathsf{Win}(\mathsf{Parity}_1 \cap \mathsf{Parity}_2)$. Assume that σ_1 is winning for the objective $(\mathsf{Feas}(\mathcal{G}) \cup \mathsf{Parity}_1) \cap \mathsf{Parity}_2$. Then for each path $\pi \in \mathsf{out}(\sigma_1)$, the following holds:*

$$\pi \in \mathsf{out}(\sigma_1) \setminus S^* X S^\omega \implies \pi \in \mathsf{Feas} \cap \mathsf{Parity}_2 \ .$$

Proof. Let $\pi = h \cdot s \cdot \pi'$ be a play in $\mathsf{out}(\sigma_1) \setminus S^* X S^\omega$. Note that s cannot be in X, otherwise π reaches X.

We first show that π is in Feas. Assume toward a contradiction that $\mathsf{cost}(h \cdot s) < 0$, since $s \notin X$, for any strategy σ_1' from s we have $\mathsf{out}(\sigma_1') \notin \mathsf{Parity}_1 \cap \mathsf{Parity}_2$. Since $\mathsf{Parity}_1 \cap \mathsf{Parity}_2$ is prefix-independent, it follows that $\pi \notin \mathsf{Parity}_1 \cap \mathsf{Parity}_2$, thus $\pi \notin (\mathsf{Feas} \cup \mathsf{Parity}_1) \cap \mathsf{Parity}_2$ a contradiction. Hence necessarily $\mathsf{cost}(h \cdot s) \geq 0$. Since this holds for any prefix $h \cdot s$ of π it follows that π is in Feas.

The fact that $\pi \in \mathsf{Parity}_2$ follows from the assumption that σ_1 is winning for Parity_2 also. ◻

We now present a construction that reduces FPP games to Feas ∩ Parity games. Before establishing the formal details of our construction, consider Figure 5 to build some intuition.

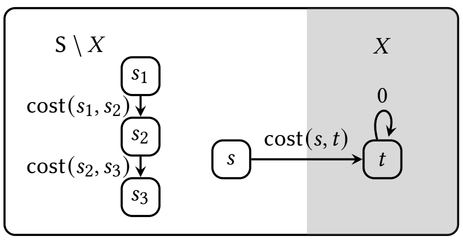

Figure 5: Illustration of the game $\hat{\mathcal{G}}$ arena construction. X is as in Lemma 4.

Let $\mathcal{G} = \langle \mathsf{S}, (\mathsf{S}_1 \uplus \mathsf{S}_2), s_{\mathsf{ini}}, \mathsf{E}\rangle$ be a two-player zero-sum game where the objective is given by Equation (9) above. We then build a fresh two-player game $\hat{\mathcal{G}} = (\mathsf{S}, \mathsf{S}_1 \uplus \mathsf{S}_2, s_{\mathsf{ini}}, \hat{\mathsf{E}})$. The construction builds the arena schematically depicted in Figure 5. Basically, the set of state is preserved, the edges in $\mathsf{S} \setminus X$ are also preserved but edges in X^2 are erased and all replaced by self loop with cost 0.

The game $\hat{\mathcal{G}}$ is formally designed as follows:

- The sets of states is S, S_1 and S_2 are unchanged and the initial state are the same.
- The set of edges $\hat{\mathsf{E}}$ is given by $\hat{\mathsf{E}}_1 \cup \hat{\mathsf{E}}_2$ where:

$$\hat{\mathsf{E}}_1 = \{(s_1, s_2) \in \mathsf{E} \mid s_1 \notin \mathsf{Win}(\mathsf{Parity}_1 \cap \mathsf{Parity}_2)\} \ ,$$
$$\hat{\mathsf{E}}_2 = \{(s, s) \mid s \in \mathsf{Win}(\mathsf{Parity}_1 \cap \mathsf{Parity}_2)\} \ .$$

- The new priority function over $\hat{\mathsf{S}}$ is obtained as follows:

$$\widehat{\mathsf{prty}}(s) = \begin{cases} 0 & \text{if } s \in \mathsf{Win}(\mathsf{Parity}_1 \cap \mathsf{Parity}_2) \ , \\ \mathsf{prty}(s) & \text{otherwise.} \end{cases}$$

- The new cost function over $\hat{\mathsf{E}}$ is defined as follows:

$$\widehat{\mathsf{cost}}(s_1, s_2) = \begin{cases} 0 & \text{when } (s_1, s_2) \in \hat{\mathsf{E}}_2 \\ \mathsf{cost}(s_1, s_2) & \text{otherwise.} \end{cases}$$

We will use the above construction to solve FPP games. We first state a couple of facts (X as in Lemma 4):

Fact 1 For any pair of states (s, t) in $(\mathsf{S} \setminus X)^2$ the priority of s and the cost of (s, t) are similar in $\mathcal{G}$ and $\hat{\mathcal{G}}$.

Fact 2 Histories and plays in $\mathcal{G}$ that never visit X are preserved in $\hat{\mathcal{G}}$. Moreover, if such a play in $\mathcal{G}$ is in $\mathsf{Feas} \cap \mathsf{Parity}_2$ then it is in Feas ∩ Parity induced by $\widehat{\mathsf{cost}}$ and $\widehat{\mathsf{prty}}$.

We want to show that the above construction preserves winning strategies in the original game $\mathcal{G}$. The preservation of strategies is formalized in the following sense:

Proposition 5. *Let $\mathcal{G}$ be an FPP game, then the following assertions are equivalent:*

a. Player 1 wins $\mathcal{G}$ (from the initial state),
b. Player 1 wins $\hat{\mathcal{G}}$ (from the initial state) for the objective Feas ∩ Parity *induced by* $\widehat{\mathsf{cost}}$ *and* $\widehat{\mathsf{prty}}$.

Proof. In the following, X is as in Lemma 4. We start by showing the following implication $a. \rightarrow b.$. Let σ_1 be a winning strategy for player 1 in $\mathcal{G}$ from s_{ini}. We build a new strategy $\hat{\sigma}_1$ in $\hat{\mathcal{G}}$ using the following mapping.
Let h be a history for player 1 in $\hat{\mathcal{G}}$, then:

$$\hat{\sigma}_1(h) = \begin{cases} \sigma_1(h) & \text{if } h \notin S^* X S^* \ , \\ \mathsf{last}(h) & \text{otherwise.} \end{cases}$$

Notice that $\hat{\sigma}_1$ is well defined, since the edge relation is preserved in $\hat{\mathcal{G}}$ for any pair of states not in X, and any state in X contains a self-loop in $\hat{\mathcal{G}}$.

Let us show that $\hat{\sigma}_1$ is winning in $\hat{\mathcal{G}}$ for player 1, i.e, we show that $\mathsf{out}(\hat{\sigma}_1) \subseteq$ Feas ∩ Parity induced by $\widehat{\mathsf{cost}}$ and $\widehat{\mathsf{prty}}$. Let $\hat{\pi} \in \mathsf{out}(\hat{\sigma}_1)$ be a play in $\hat{\mathcal{G}}$ compatible with $\hat{\sigma}_1$. Then we distinguish between two cases:

i. $\hat{\pi}$ never visits X,
ii. $\hat{\pi}$ reaches X.

Assume that *i.* holds, then $\hat{\pi}$ is not in $S^* X S^\omega$ then, $\hat{\pi} \in \mathsf{out}(\sigma_1)$ and since σ_1 is winning for player 1, Lemma 4 applies, thus $\pi \in \mathsf{Feas} \cap \mathsf{Parity}_2$.

Otherwise, *ii.* holds and $\hat{\pi} = h \cdot s^\omega$ where h is the longest prefix of $\hat{\pi}$ that never visits X and $s \in X$ is the first state along $\hat{\pi}$ in X. Thanks to *Fact* 2 about the construction of $\hat{\mathcal{G}}$, the history h is well defined in $\mathcal{G}$, and even more, it is compatible with σ_1. Assume now for the sake of a contradiction, that $\hat{\pi} \notin$ Feas. This implies that there exists a prefix h' of h such that $\mathsf{cost}(h') < 0$. So, we have $h'\pi' \notin$ Feas for every π' such that $h'\pi' \in \mathsf{Play}(\mathcal{G})$.

Now since σ_1 is winning for player 1, necessarily $h'\pi' \in \mathsf{Parity}_1 \cap \mathsf{Parity}_2$ for all π' such that $h'\pi' \in \mathsf{out}(\sigma_1)$. Hence, σ_1 is winning for player 1 from $\mathsf{last}(h')$ for the objective $\mathsf{Parity}_1 \cap \mathsf{Parity}_2$, and thus $\mathsf{last}(h') \in X$, which contradicts the assumption that $h \notin S^* X S^*$. Therefore $\pi \in$ Feas, and since $\pi = h \cdot s^\omega$, $\pi \in \mathsf{Parity}_2$ in $\hat{\mathcal{G}}$.

Let us prove the other direction, i.e, $(b. \rightarrow a.)$. Let $\hat{\sigma}_1$ be a winning strategy for player 1 in $\hat{\mathcal{G}}$ from s_{ini} for the objective Feas ∩ Parity, we build a strategy σ_1 as follows, such that for any history h in $\mathcal{G}$,

$$\sigma_1(h) = \begin{cases} \hat{\sigma}_1(h) \text{ if } h \in S^* X \, S^* \ , \\ \sigma_{\mathsf{Parity}}(\mathsf{last}(h)) \text{ if } \mathsf{last}(h) \text{ is in } X \ , \end{cases}$$

where σ_{Parity} is a winning strategy from $\mathsf{last}(h)$ for $\mathsf{Parity}_1 \cap \mathsf{Parity}_2$, which exists since $\mathsf{last}(h) \in X$.

Let us show that σ_1 is a winning strategy for player 1 for the objective $(\mathsf{Feas} \cup \mathsf{Parity}_1) \cap \mathsf{Parity}_2$. Let π be a play in $\mathsf{out}(\sigma_1)$, such a play either visits X or it stays outside X, in case it reaches X let k be the first moment it reaches X, then from $\pi[k]$ the strategy σ_1 plays accordingly to σ_{Parity}, thus π is winning from $\mathsf{Parity}_1 \cap \mathsf{Parity}_2$. In the case π never visits X it follows that σ_1 plays according to $\hat{\sigma}_1$. Notice that thanks to *Fact* 1, cost and $\widehat{\mathsf{cost}}$ coincide in this part of the arena. The same is true about prty_2 and $\widehat{\mathsf{prty}}$. Therefore, since

$\hat{\sigma}_1$ is winning for $\mathsf{Feas} \cap \mathsf{Parity}$, it follows that σ_1 is winning for $\mathsf{Feas} \cap \mathsf{Parity}_2$. ◻

From the previous proposition and classical results, we also obtain a complexity upper-bound for solving FPP games.

Corollary 6. *The problem of deciding the existence of a winning strategy in an FPP game is in* NP ∩ co-NP.

Proof. Let $\mathcal{G}$ be an FPP game, and let $\widehat{\mathcal{G}}$ the game obtained by the above construction. $\widehat{\mathcal{G}}$ is linear in the size of $\mathcal{G}$. To conclude, we notice that according to [4], deciding the existence of a winning strategy in games where the objective is $\mathsf{Feas} \cap \mathsf{Parity}$ is in NP ∩ co-NP. ◻

5 THE COMPLEXITY OF COMPUTING A RESOURCE-AWARE CONTROLLER

With the previous results, we can finally establish the complexity of the problem of non-cooperative rational synthesis when the objectives of the players are induced by an LTL specification. Then, we extend it to the case where the system has a "rich specification", that is, in addition to having the objective of maintaining the resource non-negative, the controller must also ensure a qualitative objective.

5.1 The Complexity of the Non-Cooperative Feasible Rational Synthesis

Theorem 7. *The non-cooperative feasible rational synthesis problem is* 2EXPTIME*-complete.*

Proof. The hardness easily follows from classical LTL synthesis problem [10]. The membership is as follows. From Proposition 2 computing a solution for the non-cooperative feasible rational synthesis amounts to solving a negotiation game where the winning condition is given by Equation 6. But solving a game with this objective according to Section 3.3, we need to solve a game where the objective is given by Equation 9 (which is an FPP game). However the arena built in Section 3.3 is double exponential in the size of the original game but has priority functions of polynomial size. Thanks to Proposition 5 we can invoke, the algorithm from [4] where they solve games where the objective is $\mathsf{Feas} \cap \mathsf{Parity}$. Finally, notice that the algorithm in [4] runs in time polynomial in the size of the input arena and exponential in the size of the priority function, thus using Remark 3 commenting the bounds of the negotiation game entails the upper bound. ◻

5.2 The Non-Cooperative Feasible Rational Synthesis with Rich Specifications

In this part we consider a multi-player arena $\mathcal{G} = \langle \mathsf{S}, (\mathsf{S}_1 \uplus \ldots \uplus \mathsf{S}_n), s_{\mathsf{ini}}, \mathsf{P}, \mathsf{E}\rangle$ with objectives $\mathsf{Obj}_1, \ldots, \mathsf{Obj}_n$ induced by LTL formulas and a cost function cost. We aim at designing a strategy σ_1 for player 1 such that against any strategies for players $2, \ldots, n$, the profile $\overline{\sigma} = \langle \sigma_1, \ldots, \sigma_n\rangle$ satisfies

$$\mathsf{out}(\overline{\sigma}) \in \mathsf{f\text{-}NE}(\mathcal{G}) \implies \mathsf{out}(\overline{\sigma}) \in \mathsf{Feas} \cap \mathsf{Obj}_1 \ . \tag{10}$$

We shall call this problem the *non-cooperative feasible rational synthesis with rich objectives*.

Theorem 8. *The non-cooperative feasible rational synthesis with rich specifications is* 2EXPTIME*-complete.*

Proof. Again the hardness follows from the classical LTL synthesis problem. To obtain the upper bound, one can build a negotiation arena for the input, and solve the negotiation with the following objective

$$\left((\widehat{\mathsf{Feas}} \cap \widehat{\mathsf{Obj}}_1) \cup \widehat{\mathsf{Obj}}_{\mathsf{D}}\right) \cap \widehat{\mathsf{Obj}}_{\mathsf{W}} \ , \tag{11}$$

where

$$\widehat{\mathsf{Obj}}_1 = \{\pi \in \widehat{\mathsf{S}}^{\omega} \mid \mathsf{proj}_1(\pi) \models \mathsf{Obj}_1\} \ .$$

We argue that his new objective can be encoded as a FPP game. Indeed, notice that Equation (11) is equivalent to

$$\big(\widehat{\mathsf{Feas}} \cap \underbrace{\widehat{\mathsf{Obj}}_1 \cap \widehat{\mathsf{Obj}}_{\mathsf{W}}}_{A}\big) \cup \big(\underbrace{\widehat{\mathsf{Obj}}_{\mathsf{D}} \cap \widehat{\mathsf{Obj}}_{\mathsf{W}}}_{B}\big)$$

finally we obtain the following set:

$$\big(\widehat{\mathsf{Feas}} \cup A\big) \cap C \ ,$$

where

$$A = \widehat{\mathsf{Obj}}_1 \cap \widehat{\mathsf{Obj}}_{\mathsf{W}} \ , \quad B = \widehat{\mathsf{Obj}}_{\mathsf{D}} \cap \widehat{\mathsf{Obj}}_{\mathsf{W}} \ , \quad C = A \cup B \ .$$

We conclude by applying the result from Proposition 6 and noticing that A, B, and C can be written as LTL formulas whose size is polynomial in the size of the input game using the same encoding from Section 3.3. ◻

6 CONCLUSION

We introduce the problem of non-cooperative rational synthesis where the controller has to ensure a quantitative objective. We establish the 2EXPTIME-completeness in the case where the objective of player 1 is given by a feasibility objective, c.f. Theorem 7. We also show that the same complexity result holds in the case where the objective of player 1 is a combination of both a feasibility specification and an LTL specification. In order to show these bounds, we introduce and solve a new class of two-player games that we call FPP games in Section 4 for which we show a membership in NP ∩ co-NP.

As future lines of work we first plan to study the more general case of having multiple resources in the system, as in [5]. Another research direction is investigating the interesting and more tractable fragments of LTL such as the $GR(1)$ fragment [2, 12]. Finer complexity results are also at reach, and a natural line for future research is to study the case where the qualitative specifications are given by ω-regular objectives.

ACKNOWLEDGMENTS

This research was partly funded by the UNIBZ grant Computations in Resource Aware Systems (CompRAS).

REFERENCES

[1] Alessandro Abate, Julian Gutierrez, Lewis Hammond, Paul Harrenstein, Marta Kwiatkowska, Muhammad Najib, Giuseppe Perelli, Thomas Steeples, and Michael J. Wooldridge. 2021. Rational verification: game-theoretic verification of multi-agent systems. *Applied Intelligence* 51, 9 (2021), 6569–6584. https://doi.org/10.1007/s10489-021-02658-y

[2] Roderick Bloem, Barbara Jobstmann, Nir Piterman, Amir Pnueli, and Yaniv Sa'ar. 2012. Synthesis of Reactive(1) designs. *J. Comput. System Sci.* 78, 3 (2012), 911–938. https://doi.org/10.1016/j.jcss.2011.08.007

[3] Patricia Bouyer, Romain Brenguier, Nicolas Markey, and Michael Ummels. 2015. Pure Nash Equilibria in Concurrent Deterministic Games. *Logical Methods in Computer Science* 11, 2 (2015). https://doi.org/10.2168/LMCS-11(2:9)2015

[4] Krishnendu Chatterjee and Laurent Doyen. 2012. Energy parity games. *Theoretical Computer Science* 458 (2012), 49–60. https://doi.org/10.1016/j.tcs.2012.07.038

[5] Rodica Condurache, Catalin Dima, Madalina Jitaru, Youssouf Oualhadj, and Nicolas Troquard. 2022. Careful Autonomous Agents in Environments With Multiple Common Resources. In *Proceedings of the Second Workshop on Agents and Robots for reliable Engineered Autonomy (EPTCS, Vol. 362)*. 3–14. https://doi.org/10.4204/EPTCS.362.3

[6] Rodica Condurache, Catalin Dima, Youssouf Oualhadj, and Nicolas Troquard. 2021. Rational Synthesis in the Commons with Careless and Careful Agents. In *20th International Conference on Autonomous Agents and Multiagent Systems, AAMAS'21*. ACM, 368–376.

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

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

[9] Dana Fisman, Orna Kupferman, and Yoad Lustig. 2010. Rational Synthesis. In *16th International Conference on Tools and Algorithms for the Construction and Analysis of Systems, TACAS'10*. 190–204. https://doi.org/10.1007/978-3-642-12002-2_16

[10] Erich Grädel, Wolfgang Thomas, and Thomas Wilke (Eds.). 2002. *Automata, Logics, and Infinite Games: A Guide to Current Research.* Lecture Notes in Computer Science, Vol. 2500. Springer. https://doi.org/10.1007/3-540-36387-4

[11] Julian Gutierrez, Aniello Murano, Giuseppe Perelli, Sasha Rubin, Thomas Steeples, and Michael J. Wooldridge. 2021. Equilibria for games with combined qualitative and quantitative objectives. *Acta Informatica* 58, 6 (2021), 585–610. https://doi.org/10.1007/s00236-020-00385-4

[12] Julian Gutierrez, Muhammad Najib, Giuseppe Perelli, and Michael Wooldridge. 2022. On the complexity of rational verification. *Annals of Mathematics and Artificial Intelligence* (2022). https://doi.org/10.1007/s10472-022-09804-3

[13] Orna Kupferman, Giuseppe Perelli, and Moshe Y. Vardi. 2016. Synthesis with rational environments. *Annals of Mathematics and Artificial Intelligence* 78, 1 (2016), 3–20. https://doi.org/10.1007/s10472-016-9508-8

[14] Orna Kupferman and Noam Shenwald. 2022. The Complexity of LTL Rational Synthesis. In *28th International Conference on Tools and Algorithms for the Construction and Analysis of Systems, TACAS'22*. Springer International Publishing, Cham, 25–45. https://doi.org/10.1007/978-3-030-99524-9_2

[15] John F. Nash. 1951. Non-Cooperative Games. *The Annals of Mathematics* 54 (1951), 286–295. https://doi.org/10.2307/1969529

[16] Elinor Ostrom. 1990. *Governing the Commons: The evolution of institutions for collective action.* Cambridge University Press.

[17] Amir Pnueli and Roni Rosner. 1989. On the Synthesis of a Reactive Module. In *16th Annual ACM Symposium on Principles of Programming Languages, POPL'1989*. 179–190.

[18] Thomas Steeples, Julian Gutierrez, and Michael J. Wooldridge. 2021. Mean-Payoff Games with ω-Regular Specifications. In *20th International Conference on Autonomous Agents and Multiagent Systems, AAMAS'21*. ACM, 1272–1280.