Repairing Multi-Player Games

Shaull Almagor[1], Guy Avni[1], and Orna Kupferman[1]

1 The Hebrew University, Israel.

Abstract

Synthesis is the automated construction of systems from their specifications. Modern systems often consist of interacting components, each having its own objective. The interaction among the components is modeled by a *multi-player game*. Strategies of the components induce a trace in the game, and the objective of each component is to force the game into a trace that satisfies its specification. This is modeled by augmenting the game with ω-regular winning conditions. Unlike traditional synthesis games, which are zero-sum, here the objectives of the components do not necessarily contradict each other. Accordingly, typical questions about these games concern their stability — whether the players reach an equilibrium, and their social welfare — maximizing the set of (possibly weighted) specifications that are satisfied.

We introduce and study *repair* of multi-player games. Given a game, we study the possibility of modifying the objectives of the players in order to obtain stability or to improve the social welfare. Specifically, we solve the problem of modifying the winning conditions in a given concurrent multi-player game in a way that guarantees the existence of a *Nash equilibrium*. Each modification has a value, reflecting both the cost of strengthening or weakening the underlying specifications, as well as the benefit of satisfying specifications in the obtained equilibrium. We seek optimal modifications, and we study the problem for various ω-regular objectives and various cost and benefit functions. We analyze the complexity of the problem in the general setting as well as in one with a fixed number of players. We also study two additional types of repair, namely redirection of transitions and control of a subset of the players.

1998 ACM Subject Classification J.4 Social and Behavioral Sciences, F.1.2 Modes of Computation

Keywords and phrases Nash Equilibrium, Concurrent games, Repair

1 Introduction

Synthesis is the automated construction of systems from their specifications [19]. Modern systems often consist of interacting components, each having its own objective. The interaction among the components is modeled by a *multi-player game*. Each player in the game corresponds to a component in the interaction. In each round of the game, each of the players chooses an action, and the next vertex of the game depends on the current vertex and the vector of actions chosen. A strategy for a player is then a mapping from the history of the game so far to her next action.

The strategies of the players induce a trace in the game, and the goal of each player is to direct the game into a trace that satisfies her specification. This is modeled by augmenting the game with ω-regular winning conditions, describing the objectives of the players. Unlike traditional synthesis games, which are zero-sum, here the objectives of the players do not necessarily contradict each other. Accordingly, typical questions about these games concern their stability — whether the players reach an equilibrium, and their social welfare — maximizing the set of (possibly weighted) specifications that are satisfied [23].

Different types of games can model different schemes of interaction among the components. In particular, we distinguish between *turn-based* and *concurrent* games. In the first, a single player chooses an action and determines the successor vertex in each step of the interaction. In the second, all players choose actions in all steps [1]. Another parameter is the way in which the winning conditions in the game are specified. Most common are *reachability*, *Büchi*, *co-Büchi*, and *parity* winning conditions

Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

[17], which are used to specify the set of winning traces.[1] As for stability and social welfare, here too, several types have been suggested and studied. The most common criterion for stability is the existence of a *Nash equilibrium* (NE) [18]. A profile of strategies, one for each player, is an NE if no (single) player can benefit from unilaterally changing her strategy. In the general setting of game theory, the outcome of a game fixes a reward to each of the players, thus "benefiting" stands for increasing the reward. In our setting here, the objective of a player is to satisfy her specification. Accordingly, "benefiting" amounts to moving from the set of losers – those players whose specifications are not satisfied, to the set of winners – those whose specifications are satisfied.

In [7, 22], the authors study the existence of an NE in games with Borel objectives. It turns out that while a turn-based game always has an NE [7, 22], this is not the case for concurrent games [2]. The problem of deciding whether a given concurrent game has an NE can be solved in polynomial time for Büchi games, but is NP-complete for reachability and co-Büchi games. Interestingly, this is one of the few examples in which reasoning about the Büchi acceptance condition is easier than reasoning about co-Büchi and reachability. The above results hold for a model with *nondeterministic* transition functions and with *imperfect monitoring*, where the players can observe the outcome of each transition and the vertex in which the game is, but cannot observe the actions taken by the other players [21]. In Remark 2.1 we elaborate on the difference between the two models. As we show in the paper, the results for reachability, co-Büchi, and Büchi stay valid also for our full-information model. For the parity condition, however, our model simplifies the setting and the problem of deciding the existence of an NE is NP-complete, as opposed to $\mathrm{P}_{\parallel}^{\mathrm{NP}}$ in the nondeterministic model with imperfect monitoring.

We introduce and study *repair* of multi-player games. We consider a setting with an authority (the designer) that aims to stabilize the interaction among the components and to increase the social welfare. In standard reactive synthesis [19], there are various ways to cope with situations when a specification is not realizable. Obviously, the specification has to be weakened, and this can be done either by adding assumptions on the behavior of the environment, fairness included, or by giving up some of the requirements on the system [6, 15]. In our setting, where the goal is to obtain stability, and the game is not zero-sum, a repair may both weaken and strengthen the specifications, which, in our main model, is modeled by modifications to the winning conditions.

The input to the *specification-repair problem* (SR problem, for short) is a game along with a *cost function*, describing the cost of each repair. For example, in Büchi games the cost function specifies, for each vertex v and player i, the cost of making v accepting for Player i and the cost of making v rejecting. The cost may be 0, reflecting the fact that v is accepting or rejecting in the original specification of Player i, or it may be ∞, reflecting the fact that the original classification of v is a feature of the specification that the designer is not allowed to modify. We consider some useful classes of cost functions, like *uniform costs* – where all assignments cost 1, except for one that has cost 0 and stands for the original classification of the vertex, or *don't-care costs* – where several assignments have cost 0, reflecting a don't-care original classification, and all other assignments have cost ∞. In reachability, Büchi, and co-Büchi games, we also refer to one-way costs, where repair may only add or only remove vertices from the set of accepting vertices.

The goal of the designer is to suggest a repair to the winning conditions with which the game has an NE. One way to quantify the quality of a repair is its cost, and indeed the problem also gets as input a bound on the budget that can be used in the repairs. Another way, which has to do with the social welfare, considers the specifications that are satisfied in the obtained NE. Specifically, in the *rewarded specification-repair problem* (RSR problem, for short), the input also includes a *reward function* that maps subsets of specifications to rewards. When the suggested repair leads to an NE with a set W of winners, the designer gets a reward that corresponds to the specifications of the players in W. The quality of a solution then refers both to the budget it requires and to its reward. In particular, a reward

[1] A game may also involve incomplete information or stochastic transitions or strategies. The setting we consider here is not stochastic and players have full observability on the other players actions.

function may prioritize the players and, in particular, give a reward only to one player. Then, the question of finding an NE is similar to that of *rational synthesis*, where a winning strategy for the system can take into an account the objectives of the players that constitute the environment [9]. Thus, a special case of our contribution is repair of specifications in rational synthesis.

In [4], Brenguier describes several examples in which concurrent games and their stability model real-life scenarios. This includes peer-to-peer networks, wireless channel with a shared access, shared file systems, and more. The examples there also demonstrate the practicality of specification repair in these scenarios. We give an explicit example below.

▶ **Example 1.** Consider a file-sharing system serving two users. Each user requests a file from the other user or from a repository. Accessing the repository takes longer than transmitting between users, but the connection between the users can be used only in one direction at a time. If both users request the file from each other, they each choose a bit, and the file is transmitted to one of them according to the XOR of the bits. We model the interaction between the players as a reachability game, depicted in Fig. 1.

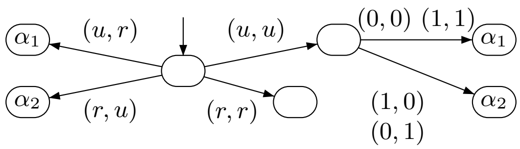

Figure 1 File sharing game. Initially, each player chooses to request either from the other user (action u) or from the repository (action r). In case both players choose u, the XOR game is played. The objective of Player i is to reach a vertex labeled α_i, in which case the other player sends her the file.

Observe that the game has no NE. Indeed, if w.l.o.g Player 1 does not reach α_1, then either Player 2 chose u and Player 1 lost in the XOR game, in which case Player 1 can deviate by choosing a different bit in the XOR game, or Player 2 chose r, in which case Player 1 can deviate by playing u.

There are several ways to repair the game such that it has an NE. One is to break the symmetry between the players and make the vertex reached by playing (u, r) accepting for both players, and similarly for the vertex reached by playing (r, u). The cost involved in this repair corresponds to the cost of communicating with the slower repository, and it is particularly useful when the reward function gives a priority to one of the players. Another possibility is to make the vertex reachable by playing (r, r) accepting for both players. Again, this involves a cost. ◀

Studying the SR and RSR problems, we distinguish between several classes, characterized by the type of winning conditions, cost functions, and reward functions. From a complexity point of view, we also distinguish between the case where the number of players is arbitrary and the one where it is constant. Recall that the problem of deciding whether an NE exists with an arbitrary number of players is NP-complete for reachability, co-Büchi, and parity games and can be solved in polynomial time for Büchi games. It is not too hard to lift the NP lower bound to the SR and RSR problems. The main challenge is the Büchi case, where one should find the cases where the polynomial complexity of deciding whether an NE exists can be lifted to the SR and RSR problems, and the cases where the need to find a repair shifts the complexity of the problem to NP. We show that the polynomial complexity can be maintained for don't-care costs, but the other settings are NP-complete. Our lower bounds make use of the fact that the unilateral change of a strategy that is examined in an NE can be linked to a change of the XOR of votes of all players, thus a single player can control the target of such transitions in a concurrent game.[2] We continue to study a setting with an arbitrary number of players. We check

[2] We note that while the representation of our games is big, as the transition function specifies all vectors of actions, our

whether fixing the number of players can reduce the complexity of the SR and RSR problems, either by analyzing the complexity of the algorithms for an arbitrary number of players, or by introducing new algorithms. We show that in many cases, we can solve the problem in polynomial time, mainly thanks to the fact that it is possible to go over all possible subsets of players in search for a subset that can win in an NE.

In the context of verification, researchers have studied also other types of repairs (c.f., [11]). After focusing on the SR and RSR problems, we turn to study two other repair models. The first is *transition-repair*, in which a repair amounts to redirecting some of the transitions in the game. As with the SR problem, each redirection has a cost, and we seek repairs of minimal cost that would guarantee the existence of an NE. The transition-repair model is suitable in settings where the actions of the players do not induce a single successor state and we can choose between several alternatives. The second model we consider is that of *controlled-players*, in which we are allowed to dictate a strategy for some players. Also here, controlling players has a cost, and we want to minimize the cost and still guarantee the existence of an NE. We study several classes of the two types, and show that they are at least as difficult as specification repair.

Due to lack of space, most proofs appear in the appendix.

2 Preliminaries

2.1 Concurrent games

A *concurrent game* is a tuple $\mathcal{G} = \langle \Omega, V, A, v_0, \delta, \{\alpha_i\}_{i\in\Omega}\rangle$, where Ω is a set of k players; V is a set of vertices; A is a set of actions, partitioned into sets A_i of actions for Player i, for $i \in \Omega$; $v_0 \in V$ is an initial vertex; $\delta : V \times A_1 \times A_2 \times \cdots \times A_k \to V$ is a transition function, mapping a vertex and actions taken by the players to a successor vertex; and α_i, for $i \in \Omega$, specifies the objective for Player i. We describe several types of objectives in the sequel. For $v, v' \in V$ and $\overline{a} \in A_1 \times A_2 \times \cdots \times A_k$ with $\delta(v, \overline{a}) = v'$, we sometimes refer to $\langle v, \overline{a}, v'\rangle$ as a transition in $\mathcal{G}$.

A *strategy* for Player i is a function $\pi_i : (A_1 \times ... \times A_k)^* \to A_i$, which directs Player i which action to take, given the history of the game so far. Note that the history is given by means of the sequence of actions taken by all players so far.[3]

A *profile* is a tuple $P = \langle \pi_1, ..., \pi_k\rangle$ of strategies, one for each player. The profile P induces a sequence $\overline{a}_0, \overline{a}_1, \ldots \in (A_1 \times ... \times A_k)^\omega$ as follows: $\overline{a}_0 = \langle \pi_1(\epsilon), ..., \pi_k(\epsilon)\rangle$ and for every $i > 0$ we have $\overline{a}_i = \langle \pi_1(\overline{a}_0, ..., \overline{a}_{i-1}), ..., \pi_k(\overline{a}_0, ..., \overline{a}_{i-1})\rangle$. For a profile P we define its *outcome* $\tau = outcome(P) \in V^\omega$ to be the path of vertices in $\mathcal{G}$ that is taken when all the players follow their strategies in P. Formally, $\tau = v_0, v_1, ...$ starts in v_0 and proceeds according to δ, thus $v_{i+1} = \delta(v_i, \overline{a}_i)$. The set of *winners* in P, denoted $W(P) \subseteq \Omega$, is the set of players whose objective is satisfied in $outcome(P)$. The set of *losers* in P, denote $L(P)$, is then $\Omega \setminus W(P)$, namely the set of players whose objective is not satisfied in $outcome(P)$.

A profile $P = \langle \pi_1, ..., \pi_k\rangle$ is a *Nash equilibrium* (NE, for short) if, intuitively, no (single) player can benefit from unilaterally changing her strategy. In the general setting, the outcome of P associates a reward with each of the players, thus "benefiting" stands for increasing the reward. In our setting here, the objective of Player i is binary – either α_i is satisfied or not. Accordingly, "benefiting" amounts to moving from the set of losers to the set of winners. Formally, for $i \in \Omega$ and some strategy π'_i for Player i, let $P[i \leftarrow \pi'_i] = \langle \pi_1, \ldots, \pi_{i-1}, \pi'_i, \pi_{i+1}, ..., \pi_k\rangle$ be the profile in which Player i *deviates* to the

complexity results hold also for games with a succinct representation of the transition function, in particular games with an arbitrary number of players in which only a constant number of players proceed in each vertex. The complexity of finding NE in succinctly represented games was studied in [10]. Succinctly represented games were studied in [16] the context of ATL model checking.

[3] Note that strategies observe the history of actions, rather than the history of vertices. In Remark 2.1 we elaborate on this aspect.

strategy π'_i. We say that P is an NE if for every $i \in \Omega$, if $i \in L(P)$, then for every strategy π'_i we have $i \in L(P[i \leftarrow \pi'_i])$.

We consider the following types of objectives. Let $\tau \in V^\omega$ be an infinite path.

- In *reachability* games, $\alpha_i \subseteq V$, and τ satisfies α_i if τ reaches α_i.
- In *Büchi* games, $\alpha_i \subseteq V$, and τ satisfies α_i if τ visits α_i infinitely often.
- In *co-Büchi* games, $\alpha_i \subseteq V$, and τ satisfies α_i if τ visits $V \setminus \alpha_i$ only finitely often.
- In *parity* games, $\alpha_i : V \to \{1, ..., d\}$, for the *index* d of the game, and τ satisfies α_i if the maximal rank that is visited by τ infinitely often is even. Formally, let $\tau = v_0, v_1, ...$, then τ satisfies α_i if $\max\{j \in \{1, ..., d\} : \alpha_i(v_l) = j$ for infinitely many $l \geq 0\}$ is even.

Note that Büchi and co-Büchi games are special cases of parity games, with ranks $\{1, 2\}$ and $\{2, 3\}$, respectively. We sometimes refer to a winning condition $\alpha_i \subseteq V$ also as a function $\alpha_i : V \to \{\top, \bot\}$, with $\alpha_i(v) = \top$ iff $v \in \alpha_i$.

▶ Remark. Our definition of strategy is based on the history of actions played. This is different from the setting in [4], where strategies are based on the history of visited vertices. Our setting reflects the fact that players have full knowledge of the actions played by other players, and not only the outcome of these actions. As we now demonstrate, our setting is different as it enables the players to make use of this full knowledge to obtain an NE. In Section 2.4 we elaborate on the algorithmic differences between the settings.

Consider the concurrent three-player Büchi game $\mathcal{G} = \langle \Omega, V, A, v_0, \delta, \{\alpha_i\}_{i \in \Omega} \rangle$, where $\Omega = \{1, 2, 3\}$, $V = \{v_0, v_1, a, b, c\}$, $A_i = \{0, 1\}$ for $i \in \Omega$, $\alpha_1 = \{a\}$, $\alpha_2 = \{b\}$, $\alpha_3 = \{c\}$, and the transition function is as follows. In v_0, if 1 and 2 play $(0, 0)$, the game moves to c, and otherwise to v_1. In v_1, Player 3 can choose to go to a or to b. The vertices a, b, and c are sinks.

There is an NE in $\mathcal{G}$, whose outcome is the path v_0, c^ω. That is, players 1 and 2 play $(0, 0)$. However, in order for this to be an NE profile, Player 3 needs to be able to "punish" either Player 1 or Player 2 if they deviate to v_1. For that, Player 3 needs to know the action that leads to v_1: if Player 1 deviates, then Player 3 chooses to proceed to b, and if Player 2 deviates, then Player 3 chooses to proceed to a. If we consider strategies that refer to histories of vertices, then there is no NE in the game. ◀

2.2 Partial games with costs and rewards

Let $\mathbb{N}_\infty = \mathbb{N} \cup \{\infty\}$. A *partial concurrent parity game* $\mathcal{G}$ is a concurrent parity game in which the winning conditions are replaced by a cost function that describes the cost of augmenting $\mathcal{G}$ with different winning conditions. Formally, $\mathcal{G} = \langle \Omega, V, A, v_0, \delta, \mathit{cost} \rangle$, where the cost function $\mathit{cost} : V \times \Omega \times \{1, ..., d\} \to \mathbb{N}_\infty$ states, for each vertex $v \in V$, player $i \in \Omega$, and rank $j \in \{1, ..., d\}$, what the cost of setting $\alpha_i(v)$ to be j. We can think of a concrete game (one with fully specified winning conditions α_i, for $i \in \Omega$) as a partial game in which the cost function is such that $\mathit{cost}(v, i, j) = 0$ if $\alpha_i(v) = j$ and $\mathit{cost}(v, i, j) = \infty$ otherwise. Intuitively, leaving $\alpha_i(v)$ as specified is free of charge, and changing $\alpha_i(v)$ is impossible, as it costs ∞. Partial games enable us to model settings where a designer can play with the definition of the winning conditions, subject to some cost function and a given budget.

Consider a partial parity game $\mathcal{G}$. A *winning-condition assignment* for $\mathcal{G}$ is $f : V \times \Omega \to \{1, ..., d\}$. The parity game induced by $\mathcal{G}$ and f, denoted $\mathcal{G}^f$, has $\alpha_i(v) = f(v, i)$, for all $v \in V$ and $i \in \Omega$. The *cost* of f is $\mathit{cost}(f) = \sum_{i \in \Omega} \sum_{v \in V} \mathit{cost}(v, i, f(v, i))$.

In the case of reachability, Büchi, and co-Büchi, the cost function is $\mathit{cost} : V \times \Omega \times \{\top, \bot\} \to \mathbb{N}_\infty$, and the winning-condition assignment is of the form $f : V \times \Omega \to \{\top, \bot\}$.

Consider a partial game $\mathcal{G}$ and a cost function cost. For every player $i \in \Omega$ and vertex $v \in V$, we define the set $\mathit{free}_{\mathit{cost}}(v, i) \subseteq \{1, ..., d\}$ as the set of ranks we can assign to $\alpha_i(v)$ free of charge. Formally, $\mathit{free}_{\mathit{cost}}(v, i) = \{j : \mathit{cost}(v, i, j) = 0\}$. We consider the following two classes of cost functions in parity games.

- *Uniform costs:* For every $i \in \Omega$ and $v \in V$, we have $|\mathit{free}_{cost}(v,i)| = 1$ and for every $j \notin \mathit{free}_{cost}(v,i)$, we have $cost(v,i,j) = 1$. Thus, a partial game with a uniform cost function corresponds to a concrete game in which we can modify the winning condition with a uniform cost of 1 for each modification.
- *Don't cares:* For every $i \in \Omega$, $v \in V$, and $j \in \{1,...,d\}$, we have $cost(v,i,j) \in \{0,\infty\}$, and $|\mathit{free}_{cost}(v,i)| \geq 1$. Thus, as in concrete games, we cannot modify the rank of vertices that are not in $\mathit{free}_{cost}(v,i)$, but unlike concrete games, here $\mathit{free}_{cost}(v,i)$ need not be a singleton, reflecting a situation with "don't cares", where a designer can choose among several possible ranks free of charge.

For the special case of reachability, Büchi, and co-Büchi games, we also consider the following classes.

- *Negative one-way costs:* For every $i \in \Omega$ and $v \in V$, either $cost(v,i,\top) = 0$ and $cost(v,i,\bot) = 1$, or $cost(v,i,\bot) = 0$ and $cost(v,i,\top) = \infty$. Intuitively, we are allowed only to modify $\top$ vertices to $\bot$ ones, thus we are only allowed to make satisfaction harder by removing vertices from α_i.
- *Positive One-way costs:* For every $i \in \Omega$ and $v \in V$, either $cost(v,i,\top) = 0$ and $cost(v,i,\bot) = \infty$, or $cost(v,i,\bot) = 0$ and $cost(v,i,\top) = 1$. Intuitively, we are allowed only to modify $\bot$ vertices to $\top$ ones, thus we are only allowed to make satisfaction easier by adding vertices to α_i.

Reward function Consider a game $\mathcal{G}$. A *reward function* for $\mathcal{G}$ is $\zeta : 2^{\Omega} \to \mathbb{N}$. Intuitively, if the players follow a profile P of strategies, then the reward to the designer is $\zeta(W(P))$. Thus, a designer has an incentive to suggest to the players a stable profile of strategies that maximizes her reward. We assume that ζ is monotone w.r.t. containment.

2.3 The specification repair problem

Given a partial game $\mathcal{G}$ and a threshold $p \in \mathbb{N}$, the *specification-repair problem* (SR problem, for short) is to find a winning-condition assignment f such that $cost(f) \leq p$ and $\mathcal{G}^f$ has an NE. Thus, we are willing to invest at most p in order to be able to suggest to the players a stable profile of strategies.

In the *rewarded specification-repair problem* (RSR problem, for short) we are also given a reward function ζ and a threshold q, and the goal is to find a winning-condition assignment f such that $cost(f) \leq p$ and $\mathcal{G}^f$ has an NE with a winning set of players W for which $\zeta(W) \geq q$.

▶ Remark. An alternative definition to the RSR problem would have required all NEs in $\mathcal{G}^f$ to have a reward greater than q. This is similar to the cooperative vs. non-cooperative definitions of rational synthesis [9, 14]. In the cooperative setting, which we follow here, we assume that the authority can suggest a profile of strategies to the players, and if this profile is an NE, then they would follow it. In the non-cooperative one, the authority cannot count on the players to follow its suggested profile even if it is an NE. We find the cooperative setting more realistic, especially in the context of repairs, which assumes rational cooperative agents (indeed, they are willing to apply a repair for a cost). Moreover, all existing work in Algorithmic Game Theory follow the cooperative setting in games that are similar to the ones we study.

Also, rather than including in the input to the RSR problem two thresholds, one could require that $\zeta(W) \geq cost(f)$ or to compare $\zeta(W)$ with $cost(f)$ in some other way. Our results hold also for such definitions. ◀

We distinguish between several classes of the SR and RSR problems, characterized by the type of winning conditions, cost function, and reward function. From a complexity point of view, we also distinguish between the case where the number of players is arbitrary and the one where it is constant.

▶ Remark. Another complexity issue has to do with the size of the representation of the game. Recall that, specifying $\mathcal{G}$, we need to specify the transition $\delta(v,\overline{a})$ for every vertex $v \in V$ and action vector $\overline{a} \in A^{|\Omega|}$. Thus, the description of $\mathcal{G}$ is exponential in the size of Ω. While this may make the lower bounds more challenging, it may also makes polynomial upper bounds easy. In Remark 3.1 we argue that our complexity results hold also in a settings with a succinct representation of $\mathcal{G}$. For example, when $\mathcal{G}$ is *c-concurrent* for some $c \geq 1$, meaning that in each vertex, only c players *control* the vertex. That is, in each vertex only c players choose actions and determine the successor vertex. Then, the size of δ is bounded by $|V \times A^c|$, for a constant c. ◀

2.4 Deciding the existence of an NE.

The problem of deciding the existence of an NE, which is strongly related to the SR problem was studied in [4]. The model there subsumes our model. First, as discussed in Remark 2.1, our strategies have full knowledge of actions, whereas the strategies in [4] only observe vertices. Second, the transition function in [4] is *nondeterministic*, thus a vertex and a vector of actions are mapped to a set of possible successors. We can efficiently convert a game in our model into an "equivalent" game in the model of [4] (in the sense that the existence of a NE is preserved). Thus, algorithmic upper bounds from [4] apply to our setting as well. Conversely, however, lower bounds from [4] do not apply to our model, and indeed the lower bounds we show differ from those of [4].

Specifically, it is shown in [4, 3] that the problem of deciding whether a given game has an NE is $\mathrm{P}_{\|}^{\mathrm{NP}}$-complete for parity objectives; that is, it can be solved in polynomial time with parallel queries to an NP oracle. The problem is NP-complete for reachability and co-Büchi objectives, and can be solved in polynomial time for Büchi games. We show that in our model, while the complexity of the problem for reachability, Büchi, and co-Büchi objectives coincides with that of [4], the complexity for parity objectives is NP-complete. In Section 3.1 we present Theorem 5, which entails an explicit algorithm for deciding the existence of an NE in Büchi games. Our algorithm is significantly simpler than the one in [4] as it considers a deterministic model.

Additionally, we emphasize that the main contribution of this work is the introduction of repairs, and our choice of model is in part for its clarity. Indeed, repair can similarly be defined in the model of [4], as partial observation is an orthogonal notion.

In Appendix A.1 we prove the following theorem.

▶ **Theorem 2.** *The problem of deciding whether a concurrent reachability, co-Büchi, or parity game has an NE is NP-complete.*

In particular, we note that the problem of *verifying* the existence of an NE can be solved in polynomial time, using an appropriate witness. See Appendix A.1 for details.

3 Solving the SR and RSR Problem

3.1 An Arbitrary Number of Players

In this section we consider the SR problem for an arbitrary number of players. Recall that the problem of deciding whether an NE exists is NP-hard for reachability, co-Büchi, and parity games and is in P for Büchi games. It is not too hard to lift the NP lower bound to the SR problem. The main challenge is the Büchi case, where one should find the cases where the polynomial complexity of deciding whether an NE exists can be lifted to the SR problem, and the cases where the need to find a repair shifts the complexity of the problem to NP.

▶ **Theorem 3.** *The SR problem for reachability, co-Büchi, and parity games with uniform, don't cares, positive one-way, or negative one-way costs is NP-complete.*

Proof. Membership in NP is easy, as given a game $\mathcal{G}$, a cost function $cost$, and a threshold p, we can guess a winning-condition assignment f, and then proceed to nondeterministically check whether there exists an NE in $\mathcal{G}^f$ as described in Section 2.4.

For the lower bound, we describe a reduction from the problem of deciding whether an NE exists in a given co-Büchi or reachability game is NP-complete, proved to be NP-hard in Theorem 2.

Consider a game $\mathcal{G}$, and let $cost$ be the cost function induced naturally by it. That is, for every $v \in V$ and $i \in \Omega$, we have $cost(v, i, \alpha_i(v)) = 0$, and the rest of the cost function is defined to involve a positive cost and respect the definition of uniform, don't cares, positive one-way, or negative one-way cost function. With this cost function, the only assignment with cost 0 is such that $f(v, i) = \alpha_i(v)$ for every $i \in \Omega$ and $v \in V$. Thus, $\mathcal{G}$ has an NE iff there is a winning-condition assignment f such that $cost(f) \leq 0$ and $\mathcal{G}^f$ has an NE. ◀

We turn to Büchi games, where the goal is to find the cases where the polynomial complexity of deciding the existence of an NE can be maintained. We start with the negative cases.

▶ **Theorem 4.** *The SR problem for Büchi games with uniform, positive one-way, or negative one-way costs is NP-complete.*

Proof. Membership in NP is easy, as given a game $\mathcal{G}$, a cost function $cost$, and a threshold p, we can guess a winning-condition assignment f, and then check in polynomial time whether there exists an NE in $\mathcal{G}^f$ [4]. For the lower bounds we describe reductions from SET-COVER, which is well known to be NP-complete [12]. We bring its definition here for completeness. Consider a set $U = \{1, \ldots, n\}$ of elements and a set $S = \{S_1, ..., S_m\}$ of subsets of U, thus $S_i \subseteq U$ for every $1 \leq j \leq m$. A set-cover of size ℓ is $\{S_{j_1}, ..., S_{j_\ell}\} \subseteq S$ such that for every $i \in U$ there exists $1 \leq l \leq \ell$ such that $i \in S_{j_l}$. The SET-COVER problem is to decide, given U, S, and ℓ, whether there exists a set-cover of size ℓ. We assume w.l.o.g that $\ell < \min\{n, m\}$.

Uniform costs. Consider an input $\langle U, S, \ell\rangle$ for SET-COVER. We construct a partial concurrent game $\mathcal{G} = \langle \Omega, V, A, v_0, \delta, cost\rangle$ such that there is a set cover of U of size ℓ iff there exists a winning-condition assignment f with $cost(f) \leq \ell$ such that $\mathcal{G}^f$ has an NE.

The players in $\mathcal{G}$ are $\Omega = U \cup S$. That is, there is one player, referred to as Player i, for every $i \in U$, and one player, referred to as Player S_j, for every $S_j \in S$. The set of vertices in $\mathcal{G}$ is $V = U \cup \{\langle S_j, i\rangle : i \in S_j \in S\} \cup \{v_{\text{end}}\}$. The initial vertex is $1 \in U$. We now describe the transitions and actions (see Fig. 2).

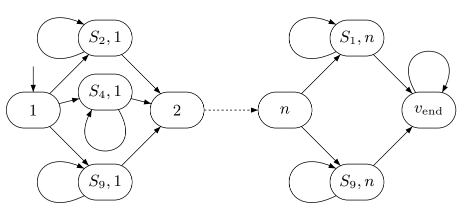

Figure 2 Reduction in the uniform costs setting in Theorem 4. Here, $1 \in S_2 \cap S_4 \cap S_9$, and $n \in S_1 \cap S_9$.

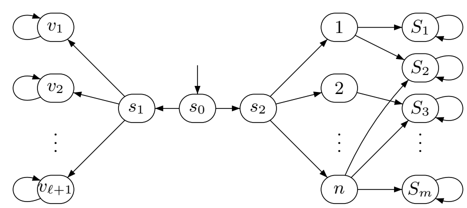

Figure 3 Reduction of the negative one-way costs setting in Theorem 4. Here, $1 \in S_1 \cap S_2$, $2 \in S_3$ and $n \in S_2 \cap S_3 \cap S_m$.

At vertex $i \in U$, Player i alone has control, in the sense that only her action is taken into an account in deciding the successor. Player i can choose to move to a vertex $\langle S_j, i\rangle$ for which $i \in S_j$. At vertex $\langle S_j, i\rangle$, all players in $U \cup \{S_j\}$ have control on the choice of the successor vertex and can choose either to stay at $\langle S_j, i\rangle$, or to proceed, either to vertex $i + 1$, if $i < n$, or to v_{end}, if $i = n$. This choice is made as follows. The actions of the players are $\{0, 1\}$, and the transition depends on the XOR of the actions. If the XOR is 0, then the game stays in $\langle S_j, i\rangle$ and if the XOR is 1, the game proceeds to $i + 1$ or to v_{end}. Finally, v_{end} has a self loop.

We now describe the cost function. Intuitively, we define $cost$ so that the default for v_{end} is to be accepting for all players $i \in U$ and rejecting for all players $S_j \in S$. Thus, $cost(v_{\text{end}}, i, \top) = 0$ for all $i \in U$ and $cost(v_{\text{end}}, S_j, \bot) = 0$ for all $S_j \in S$. Also, for every $S_j \in S$ and $i \in S_j$, we have $cost(\langle S_j, i\rangle, S_j, \top) = 0$ and $cost(\langle S_j, i\rangle, i, \bot) = 0$. Thus, $\langle S_j, i\rangle$ is accepting for Player S_j and is rejecting for Player i. All other costs are set to 1, as required by a uniform cost.

We claim that $\langle U, S, \ell \rangle \in$ SET-COVER iff $\mathcal{G}$ has a winning-condition assignment f with cost at most ℓ such that $\mathcal{G}^f$ has an NE. In Appendix A.3 we formally prove the correctness of the reduction. Intuitively, every assignment f of cost at most ℓ must set $f(v_{\text{end}}, i) = \top$ and $f(v, i) = \bot$ for $v \neq v_{\text{end}}$, for some $i \in U$. Thus, an NE must end in v_{end}, as otherwise Player i uses the XOR transitions in order to deviate to a strategy whose outcome reaches v_{end}. Hence, an assignment must set $f(v_{\text{end}}, S_j) = \top$ for at most ℓ players $S_{j_1}, ..., S_{j_\ell}$, such that it is possible to get from 1 to v_{end} by going only through vertices $\langle S_{j_k}, i \rangle$ for $1 \leq k \leq \ell$. These ℓ players induce a set cover. The other direction is easy.

Positive one-way costs. In the correctness proof of the reduction above (see Appendix A.3), we show that in fact, the only assignments that need to be considered are positive one-way. Thus, the same reduction, in fact with a simpler correctness argument, can be used to show NP-hardness of the setting with positive one-way costs.

Negative one-way costs. Finally, we consider the setting of negative one-way costs. Again, we describe a reduction from SET-COVER. Consider an input $\langle U, S, \ell \rangle$ for SET-COVER. We construct a partial two-player game $\mathcal{G} = \langle \Omega, V, A, v_0, \delta, \mathit{cost} \rangle$ such that there is a set cover of U of size ℓ iff there exists a winning-condition assignment f with $\mathit{cost}(f) \leq \ell$ such that $\mathcal{G}^f$ has an NE. The game $\mathcal{G}$ is constructed as follows. The players are $\Omega = \{1, 2\}$. The vertices are $V = U \cup S \cup \{s_0, s_1, s_2\} \cup \{v_1, ..., v_{\ell+1}\}$. The game starts in s_0, where the actions for the players are $\{0, 1\}$. If the XOR of the actions is 0, the game moves to vertex s_1, where Player 1 chooses a vertex from $v_1, ..., v_{\ell+1}$, all of which have self loops. We set $\mathit{cost}(v_i, 1, \top) = 0$ for $1 \leq i \leq \ell+1$. Intuitively, if the game proceeds to s_1, then Player 1 can choose a winning vertex, and the play gets stuck there. If the XOR in s_0 was 1, the game proceeds to vertex s_2 from which player 2 chooses a vertex $i \in U$. In vertex i, player 1 chooses a vertex S_j such that $i \in S_j$. For every $1 \leq j \leq m$, the vertex S_j has only a self loop. We set $\mathit{cost}(S_j, 2, \top) = 0$ for $1 \leq j \leq m$. Intuitively, if the game proceeds to s_2, then Player 2 "challenges" Player 1 with a value $i \in U$, and Player 1 has to respond with some set S_j such that $i \in S_j$, and then the play gets stuck in S_j. See Fig. 3 for an illustration.

The rest of the cost function is set to give $\bot$ cost 0, and is completed to be a negative one-way cost. That is, we set $\mathit{cost}(v_j, 2, \bot) = 0$ for every $1 \leq j \leq \ell + 1$, $\mathit{cost}(S_j, 2, \bot) = 0$ for every $1 \leq j \leq m$, and $\mathit{cost}(x, 1, \bot) = \mathit{cost}(x, 2, \bot) = 0$ for $x \in U \cup \{s_0, s_1, s_2\}$. Finally, $\mathit{cost}(v, i, \bot) = 1$ if $\mathit{cost}(v, i, \top) = 0$ and $\mathit{cost}(v, i, \top) = \infty$ if $\mathit{cost}(v, i, \bot) = 0$, for every $i \in \{1, 2\}$ and $v \in V$, as per the definition of a negative one-way cost.

In Appendix A.4 we formally prove the correctness of the reduction. Intuitively, every assignment f of cost at most ℓ must set $f(v_i, 1) = \top$ for some $1 \leq i \leq \ell + 1$. Thus, Player 1 is guaranteed to be able to deviate and win in any profile. In order to have an NE, we must be able to set $f(S_j, 2) = \bot$ for at most ℓ vertices $S_{j_1}, ..., S_{j_\ell}$, such that for every $i \in U$ that Player 2 chooses, there exists $1 \leq k \leq \ell$ such that $i \in S_{j_k}$, and so there is a set-cover. The other direction is again, easy. ◀

We now turn to consider the positive case, where the polynomial complexity of deciding whether an NE exists can be lifted to the SR problem.

▶ **Theorem 5.** *The SR problem for Büchi games and don't-cares can be solved in polynomial time.*

Proof. Consider a partial Büchi game $\mathcal{G} = \langle \Omega, V, A, v_0, \delta, \mathit{cost} \rangle$ with don't-cares. For every $i \in \Omega$, the set of vertices V can be partitioned into three sets:

1. The set $F_i = \{v : \mathit{cost}(v, i, \top) = 0 \wedge \mathit{cost}(v, i, \bot) = \infty\}$, of *accepting* vertices.
2. The set $R_i = \{v : \mathit{cost}(v, i, \bot) = 0 \wedge \mathit{cost}(v, i, \top) = \infty\}$, of *rejecting* vertices.
3. The set $DC_i = \{v : \mathit{cost}(v, i, \bot) = \mathit{cost}(v, i, \top) = 0\}$, of *don't-care* vertices.

The SR problem then amounts to deciding whether there is an assignment $f : \bigcup_{i \in \Omega} DC_i \to \{\top, \bot\}$ such that $\mathcal{G}^f$ has an NE. Note the cost of every such assignment is 0.

For a set $S \subseteq V$, let $W_S \subseteq \Omega$ be the set of *potential winners* in S: players that either have an accepting or don't-care vertex in S. Formally, $W_S = \{i \in \Omega : (F_i \cup DC_i) \cap S \neq \emptyset\}$. The set of *losers* in S is then $L_S = \Omega \setminus W_S$, thus $i \in L_S$ iff $S \subseteq R_i$.

We describe the intuition behind our algorithm. An outcome of a profile is an infinite path in $\mathcal{G}$, which gets stuck in a SCC S. We distinguish between the case S is an ergodic SCC – one that has no outgoing edges to other SCCs in $\mathcal{G}$, and the case S is not ergodic. Our algorithm tries to find a *witness* ergodic SCC S: one for which there is an assignment f such that $\mathcal{G}^f$ has an NE whose outcome gets stuck in S. When an ergodic SCC cannot serve as a witness, it is removed from $\mathcal{G}$ along with transitions that guarantee the soundness of such a removal, and the search for a witness ergodic SCC in the new game continues. When all SCCs are removed, the algorithm concludes that no NE exists.

In order to examine whether an ergodic SCC S can serve as a witness, the algorithm checks whether the players in W_S can force the game to reach S. Once the game reaches S, every outcome would not satisfy the objective of the players in L_S. Moreover, consider the assignment f that sets, for $i \in W_S$, every vertex in DC_i to $\top$. The profile whose outcome visits all the vertices in S is an NE in G^f. Checking whether the players W_S can force the game to reach S is not straightforward, as it should take into an account possible collaboration from players in L_S that are doomed to lose anyway and thus have no incentive to deviate from a strategy in which they collaborate with the players in W_S. In Appendix A.5 we formalize this intuition and give full details of the algorithm. Essentially, the polynomial complexity follows from the fact that we solve k zero-sum Büchi games on the structure of $\mathcal{G}$.

◀

▶ **Remark.** As discussed in Remark 2.3, our results stay valid when the games are c-concurrent for a constant $c \geq 2$. In particular, the running time of the algorithm described in Theorem 5 is polynomial in the representation size of $\mathcal{G}$. As for lower bounds, the second reduction described in the proof of Theorem 4 generates a game with only two players. In addition, the first reduction there can be slightly modified to capture 2-concurrent games. For that, we replace the vertices $S \times U$ in $\mathcal{G}$ by a cycle of n vertices, where $\langle S_j, i\rangle$ is the first vertex in the cycle $\langle S_j, i\rangle_1, ..., \langle S_j, i\rangle_n$. The players that control the l-th vertex, for $1 \leq l \leq n$, are S_j and l. Both players have two possible actions $\{0, 1\}$. If the XOR of their choices is 0, the game continues to $(l+1) \mod n$, the next vertex in the cycle, and if it is 1, then the game exits the cycle and proceeds to vertex $i+1$. Clearly, each player in $U \cup \{S_j\}$ can force the game to stay in the gadget or exit it assuming the other players fix a strategy.

3.2 A Constant Number of Players

In this section, we consider the SR problem for a constant number of players. The algorithms presented in Section 3.1 can be clearly applied in this setting. For example, Theorem 5 implies that the SR problem for Büchi games with don't cares can be solved in polynomial time, and in particular this holds when the number of players is fixed. For NP-complete problems, however, the upper bounds in Section 3.1 only imply exponential time algorithms. In this section, we check whether fixing the number of players can reduce the complexity, either by analyzing the complexity of the algorithms from Section 3.1, or by introducing new algorithms.

The results are summarized in Table 1, and the proofs appear in Appendix A.6, with the exception of Theorem 6 below.

▶ **Theorem 6.** *The SR problem for co-Büchi games with positive one-way costs and a constant number of players can be solved in polynomial time.*

Proof. We solve the problem by presenting a polynomial time algorithm for checking, given a game $\mathcal{G}$, a bound $p \in \mathbb{N}$ on the budget for the repair, and a set $W \subseteq \Omega$, whether there is a positive one-way assignment f with cost at most p, for which $\mathcal{G}^f$ has an NE profile P with $W \subseteq W(P)$. We then iterate over all subsets $W \subseteq \Omega$ to obtain a polynomial time algorithm.

Under the definitions used in the proof of Theorem 5, let $L = \Omega \setminus W$, and let $\mathcal{G}_W = \mathcal{G}|_{\text{doomed}(L)}$. Consider a vertex $v \in V$ that is reachable from v_0 in $\mathcal{G}_W$. We look for an assignment f for which there is a cycle that contains v and traverses only vertices in $\bigcap_{i \in W} \alpha_i^f$. Such a cycle satisfies the objectives

Problem ╲ Game	Büchi	co-Büchi	Reachability	Parity
NE Existence	P [3]	P [Th. 10]	P [2]	NP∩coNP [Th. 10]
Uniform	P [Th. 11]	NP-C [Th. 14]	P [Th. 14]	NP-C [Th.14]
Don't care	P [Th. 5]	P [Th. 12]	P [Th. 12]	NP∩coNP [Th. 12]
Negative One-way	NP-C [Th. 13]			–
Positive One-way	P [Th. 11]	P [Th. 6]	P [Th. 11]	–

Table 1 Complexity results for the setting with a constant number of players.

of the players in W. In order to do so, we add weights to $\mathcal{G}_W$ as follows. The weight of an edge $\langle u, u' \rangle$ in $\mathcal{G}_W$ is the repair budget that is needed in order to make u' accepting for all players in W. That is, $\langle u, u' \rangle$ gets the weight $\sum_{i \in W} cost(u', i, \top)$. Then, we run Dijkstra's shortest-path algorithm from v to find the minimal-weight cycle that contains v. If the weight of the cycle is at most p, we return "yes". If there is no such cycle for every $v \in V$ and $W \subseteq \Omega$, we return "no". We then repeat this process for every $W \subseteq \Omega$.

In Appendix A.7 we analyze the runtime and prove the correctness of the algorithm. ◀

3.3 Solving the RSR problem

Recall that in the RSR problem we are given, in addition to $\mathcal{G}$, $cost$, and $p \in \mathbb{N}$, a reward function $\zeta : 2^{\Omega} \to \mathbb{N}$ and a threshold q, and we need to decide whether we can repair $\mathcal{G}$ with cost at most p in a way that the set of winners W in the obtained NE is such that $\zeta(W) \geq q$. In Appendix A.8 we argue that the additional requirement about the reward does not change the complexity of the problem.

▶ **Theorem 7.** *The complexity of the SR and RSR problems coincide for all classes of objectives and cost functions, for both an arbitrary and a constant number of players.*

4 Other Types of Repairs

So far, we studied repairs that modify the winning conditions of the players. Other types of repairs can be considered. In this section, we examine two such types: *transition repair*, which modifies the transitions of the game, and *controlled-players repair*, where we can control (that is, force a strategy) on a subset of the players. The later is related to the *Stackelberg model*, which has been extensively studied in economics and more recently in Algorithmic Game Theory [13, 20], and in which some of the players are selfish whereas others are controllable.

4.1 Transition repair

In the *transition repair* model, we are allowed to redirect the transitions of a game. Transition repair is suitable in cases where a system is composed of several concurrent components, and we have some control on the flow of the entire composition. For example, when actions of the players correspond to assignments to variables, but the state space of the system is richer than valuations of the variables assigned by the players. So, each state of the system corresponds to the current assignments to the variables the players control as well as an additional component that the designer controls. As another, more specific example, consider a system in which several threads request a lock and granting a lock to a certain thread is modeled by a transition. Redirecting this transition can correspond to the lock being given to a different thread. Typically, not all repairs are possible, which is going to be modeled by an ∞ cost to impossible repairs. Finally, the games we study are sometimes obtained from LTL specifications of the players. Repairs in the winning conditions then have the flavor of switching between "until" and "weak-until" in the LTL specification. In this setting, one may find transition-repair to be more appropriate. First, it enables more elaborate changes in the specifications. Secondly, changes in the acceptance condition of the nondeterministic Büchi automata for the specifications induce changes of transitions in their deterministic parity automata, which compose the game.

In Appendix A.9 we formalize this model, and define the *transition-repair problem* (TR problem, for short) similarly to the SR problem, with the goal being to find a cheap transition-repair that guarantees the existence of an NE. We prove the following results.

▶ **Theorem 8.** *The TR problem is NP-complete for the following cases:*

- *A constant number of players, for all objectives.*
- *Uniform costs with an arbitrary number of players, for all objectives.*
- *Uniform costs with a constant number of players, and co-Büchi and parity objectives.*

and can be solved in polynomial time for uniform costs with a constant number of players and Büchi and reachability objectives. The TR problem with uniform costs can be solved in polynomial time for Büchi and reachability objectives, and is NP complete

4.2 Controlled-players repair

The underlying assumption in game theory is that players are selfish and rational. In particular, they would follow a suggested strategy only if it is in their interest. In the *controlled-players repair* model, we assume that we can control some of the players and guarantee they would follow the strategy we assign them. The other players cooperate only if the profile is an NE. Controlling a player has a cost and our goal is to reach such a profile with a minimal cost. This model is a type of Stackelberg model, where there is a *leader* player whose goal is to increase the social welfare. She moves first, selects a fraction α of the players, and assigns strategies to them. The rest of the players are selfish and choose strategies to maximize their revenue. Previous works in Algorithmic Game Theory study how the parameter α affects the social welfare in an NE. Clearly, when α is high, the social welfare increases.

Formally, given a game $\mathcal{G} = \langle \Omega, V, A, v_0, \delta, \{\alpha_i\}_{i\in\Omega}\rangle$ and a *control cost* function $cost : \Omega \to \mathbb{N}_\infty$, which maps each agent to the cost of controlling him, the *controlled-player repair* problem (the CR problem, for short), is to find a set of players of minimal cost such that if we are allowed to fully control these players, then the game has an NE. By *controlling* we mean that the players are not allowed to deviate from their strategies in the suggested profile.

Controlled-players repair arises in settings where an unstable system can be stabilized by restricting the environment, but this involves a cost. For example, controlling players is possible in settings where players accept an outside payment. As another example, taken from [20], the players are customers who can either pay a full price for using a system, and then their choices are unlimited, or they can pay a "bargain" price, and then their choices are limited, and hence their quality of service is not guaranteed. As a third example, consider a system that receives messages from the environment. We may want to require that messages arrive chronologically, otherwise our system is unstable. We can require this, but it involves a latency cost, and is effectively translated to asking the message dispatching thread to work in a non-optimal way, which is not the best strategy for the message dispatch server.

In the decision version of the problem, we are given a threshold p, and we need to determine if there exists a set $S \subseteq \Omega$ such that $cost(S) = \sum_{i\in S} cost(i) \leq p$ and controlling the players in S ensures the existence of an NE.

We start by studying the general case. In order to solve the CR problem, we observe that controlling Player i can be modeled by setting α_i to be the most permissive, thus for reachability, Büchi, and co-Büchi objectives, we set $\alpha_i = V$, and for parity objectives $\alpha_i(v)$ is the maximal even index. Indeed, if there is an NE profile P in $\mathcal{G}$ in which we control Player i, then P is also an NE when we set α_i as in the above (without controlling player i). Clearly, Player i has no incentive to deviate. Conversely, if there is an NE profile P after setting α_i to be the most permissive, then the same profile P is an NE in a game in which we control Player i and force it to play his strategy in P.

Theorem 9 below summarizes our results, and is proved in Appendix A.10.

▶ **Theorem 9.** *The CR problem is NP-complete for reachability, co-Büchi, and parity objectives, as well as for c-concurrent Büchi games, and is in P for general Büchi games, and for all objectives with a constant number of players.*

▶ Remark. In the future, we plan to investigate *scheduling repairs*, where a repair controls the set of players that proceed in a vertex, as well as *disabling repairs*, in which some actions of some players are disabled in some vertices.

References

1 R. Alur, T.A. Henzinger, and O. Kupferman. Alternating-time temporal logic. *Journal of the ACM*, 49(5):672–713, 2002.
2 P. Bouyer, R. Brenguier, and N. Markey. Nash equilibria for reachability objectives in multi-player timed games. In *Proc. 21st CONCUR*, pages 192–206, 2010.
3 P. Bouyer, R. Brenguier, N. Markey, and M. Ummels. Nash equilibria in concurrent games with Büchi objectives. In *Proc. 31st FSTTCS 2011*, pages 375–386, 2011.
4 R. Brenguier. Nash Equilibria in Concurrent Games – Application to Timed Games. PhD thesis, École normale supérieure, 2012.
5 K. Chatterjee, L. de Alfaro, and T.A. Henzinger. Qualitative concurrent parity games. *ACM Trans. Comput. Log.*, 12(4):28, 2011.
6 K. Chatterjee, T. Henzinger, and B. Jobstmann. Environment assumptions for synthesis. In *Proc. 19th CONCUR*, LNCS 5201, pages 147–161. Springer, 2008.
7 K. Chatterjee, R. Majumdar, and M. Jurdzinski. On nash equilibria in stochastic games. In *Proc. 13th CSL*, LNCS 3210, pages 26–40. Springer, 2004.
8 L. de Alfaro and T.A. Henzinger. Concurrent ω-regular games. In *15th LICS*, pages 141–154, 2000.
9 D. Fisman, O. Kupferman, and Y. Lustig. Rational synthesis. In *Proc. 16th TACAS*, LNCS 6015, pages 190–204. Springer, 2010.
10 G. Gottlob, G. Greco, and F. Scarcello. Pure Nash equilibria: hard and easy games. In *Proc. 9th TARK*, pages 215–230, 2003.
11 B. Jobstmann, A. Griesmayer, and R. Bloem. Program repair as a game. In *Proc. 17th CAV*, LNCS 3576, pages 226–238, Springer 2005,
12 R.M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pages 85–103, 1972.
13 A. Korilis, A. Lazar and A. Orda. Achieving Network Optima Using Stackelberg Routing Strategies. In *IEEE/ACM Trans. Netw.*, 5(1):161–173, 1997.
14 O. Kupferman, G. Perelli, and M.Y. Vardi. Synthesis with rational environments. In *Proc. 12th EUMAS*. Springer, 2014.
15 W. Li, L. Dworkin, and S. A. Seshia. Mining assumptions for synthesis. In *Proc. 9th MEMOCODE*, pages 43–50, 2011.
16 F. Laroussinie, N. Markey, and G. Oreiby. On the Expressiveness and Complexity of ATL. *LMCS*, 4(2), 2008.
17 D.A. Martin. Borel determinacy. *Annals of Mathematics*, 65:363–371, 1975.
18 J.F. Nash. Equilibrium points in n-person games. In *Proceedings of the National Academy of Sciences of the United States of America*, 1950.
19 A. Pnueli and R. Rosner. On the synthesis of a reactive module. In *Proc. 16th POPL*, 1989.
20 D. Rosenberg, E. Solan, and N. Vieille. The maxmin value of stochastic games with imperfect monitoring. *Int. J. Game Theory*, 32(1):133–150, 2003.
21 T. Roughgarden. Stackelberg Scheduling Strategies. *SIAM J. Comput.*, 33(2):332–350, 2004.
22 M. Ummels. The complexity of nash equilibria in infinite multiplayer games. In *Proc. 11th FOSSACS*, pages 20–34, 2008.
23 J. von Neumann and O. Morgenstern. Theory of games and economic behavior.

A Proofs

A.1 Verifying the existence of an NE

In this section we show that the problem of verifying the existence of NE is solvable in polynomial time, thus concluding the proof of Theorem 2.

Consider a parity game $\mathcal{G} = \langle \Omega, V, A, v_0, \delta, \{\alpha_i\}_{i\in\Omega}\rangle$.

Game against i Consider a player $i \in \Omega$. We define the *game against* i to be a two-player zero-sum concurrent game, where the players are Player i and the coalition $\Omega \setminus \{i\}$. The game is played on $\mathcal{G}$, where the objective of player i is α_i, and the objective of $\Omega \setminus \{i\}$ is to prevent i from satisfying α_i. For example, in parity games, the objective of the coalition is to generate an outcome in which the maximal index that is visited infinitely often is odd. The *cage* for player i is the set of vertices $C_i \subseteq V$ that consists of all vertices from which the coalition wins the game against i.[4] Deciding whether a vertex v is in C_i amounts to solving a two-player zero-sum concurrent game. These games can be solved in polynomial time for reachability, Büchi, and co-Büchi objectives [8] and in $\mathrm{NP} \cap \mathrm{coNP}$ for parity objectives [5].

Doomed transitions Consider a transition $t = \langle v, \overline{a}, v'\rangle$ with $v, v' \in C_i$, thus $\delta(v, \overline{a}) = v'$. We say that t is *doomed for Player* i if Player i cannot alter her action in $\overline{a}$ and escape the cage C_i. Formally, for $a'_i \in A_i$, let $\overline{a}[i \leftarrow a'_i] \in A_1 \times ... \times A_k$ be the vector of actions obtained from $\overline{a}$ by changing Player i's action to a'_i. We say that t is doomed for Player i if for every $a'_i \in A_i$, we have $\delta(v, \overline{a}[i \leftarrow a'_i]) \in C_i$. For $B \subseteq \Omega$, we denote by $\mathrm{doomed}(B)$ the set of transitions that are doomed for all players in B.

The suspicious reader may wonder if it is possible for a transition in $\langle v, \overline{a}, v'\rangle$ with $v, v' \in C_i$ not to be doomed, as it is between vertices in the cage. In Fig. 4 we demonstrate that indeed such a scenario is possible.

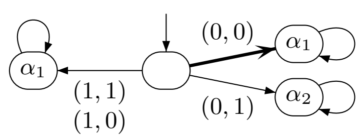

Figure 4 In the game above, the bold transition is not doomed for Player 2, even though it connects two vertices in C_2. The initial vertex is in C_2 since Player 1 can play 1, thus guaranteeing the play ends in the left vertex, which is not winning for player 2, but if the transition $(0, 0)$ is taken, then player 2 can deviate to the transition on $(0, 1)$, thus escaping the cage.

Verifying an NE Consider a profile $P = \langle \pi_1, ..., \pi_k\rangle$. Let $outcome(P) = \tau = \tau_0, \tau_1, ...$, and, for $j \geq 0$, let $\overline{a}_j$ be the action taken in the j-th transition in τ. We claim that if P is an NE, then for every $j \geq 0$, the transition $(\tau_j, \overline{a}_j, \tau_{j+1})$ is doomed for i, for every $i \in L(P)$. Indeed, since P is an NE, no player in $i \in L(P)$ has an incentive to deviate unilaterally, implying that the strategies of the players in $\Omega \setminus \{i\}$ form a winning strategy of the coalition in the game against i played from the vertex reached after the deviation. Moreover, we claim that if there is an infinite path $\tau = \tau_1, \tau_2, \ldots$ that satisfies the objectives of some set $W \subseteq \Omega$ of players and traverses only transitions that are doomed for the players in $L = \Omega \setminus W$, then there is an NE. Consider a profile P that consists of strategies whose outcome is τ. Note that since τ is a path in $\mathcal{G}$, such a profile P exists. Note also that if a player $i \in L$ deviates from the action she is expected to perform in τ, then the game reaches a vertex in the cage C_i. Thus, we set the strategies in P so that if such a deviation occurs, the coalition switches to its winning strategy. Thus, such a deviation is not beneficial for Player i, and P is an NE.

---

[4] Note that concurrent zero-sum games need not be *determined*. That is, the set of vertices is not necessarily partitioned between vertices that are winning for Player i and these that are winning for the coalition.

Finally, consider a profile P. The outcome of P is a path $\tau = \tau_0, \tau_1, ...$ that starts in v_0, and after a finite number of transitions stays forever in some strongly connected component (SCC) S. We can now use the above observation for proving membership in NP. In order to verify that a game $\mathcal{G}$ has an NE, we are given as a witness a set of vertices $S \subseteq V$ and a sequence $x_1, ..., x_k$ of vertices. We can verify in polynomial time that S is a SCC and that $x_1, ..., x_k$ is a path that leads from v_0 to S. We can also compute in polynomial time the set $W \subseteq \Omega$ of players that are winning in a profile that traverses $x_1, ..., x_k$ and then visits all vertices in S infinitely often. Finally, for every $i \in \Omega \setminus W$, we can repeatedly solve the game against i and verify in polynomial time that all the transitions through $x_1, ..., x_k$ and S are doomed for i.

Note that for parity games, there is no known polynomial algorithm to solve the games against i. We overcome this issue by guessing, for every $i \in \Omega$, the cage C_i as well as a winning memoryless strategy for the coalition. Verifying that a memoryless strategy is winning can be done in polynomial time.

Lower bounds For an NP lower bound, we show in Appendix A.2 that the hardness of the problem for reachability and co-Büchi objectives is carried over to our setting, implying a lower bound also for parity. We describe the proof for co-Büchi games, which are a special case of parity games. The proof for reachability games is similar.

A.2 The lower bound of Theorem 2

We describe a reduction from 3SAT. Consider a 3CNF formula $\varphi = c_1 \wedge ... \wedge c_m$ over the variables $x_1, ..., x_n$, where the clauses are $c_i = \ell_i^1 \vee \ell_i^2 \vee \ell_i^3$ for every $1 \leq i \leq m$ and each ℓ_i^j is a variable or its negation. We construct a co-Büchi game $\mathcal{G} = \langle \Omega, V, v_0, A, \delta, \{\alpha_i\}_{i \in \Omega} \rangle$ such that $\mathcal{G}$ has an NE iff φ is satisfiable.

The players are $\Omega = \{0, \top_1, \bot_1, \top_2, \bot_2, ..., \top_n, \bot_n\}$. Thus, we have a special player 0, as well as two players associated with each variable. Let $V_{\top,\bot} = \{\top_j, \bot_j : j \in \{1, \ldots, n\}\}$. The vertices are $V = \{1, \ldots, m\} \cup V_{\top,\bot} \cup \{\langle x_j, i \rangle, \langle \neg x_j, i \rangle : i \in \{1, \ldots, m\}$ and $j \in \{1, \ldots, n\}\}$. The initial vertex is 1.

The acceptance condition is as follows. Player 0's objective is to avoid the vertices $V_{\top,\bot}$. Thus, $\alpha_0 = V \setminus V_{\top,\bot}$. For every $1 \leq j \leq n$, Player $\top_i$'s objective is to avoid the vertices $V_{\top,\bot}$ excluding the one that belongs to her, as well as to avoid vertices of the form $\langle \neg x_j, i \rangle$, for $1 \leq i \leq m$. Player $\bot_i$'s objective is dual. Thus, for every $1 \leq j \leq n$, we have $\alpha_{\top_j} = V \setminus \big(\{\langle \neg x_j, i \rangle : i \in \{1, \ldots, m\}\} \cup (V_{\top,\bot} \setminus \{\top_j\})\big)$ and $\alpha_{\bot_j} = V \setminus \big(\{\langle x_j, i \rangle : i \in \{1, \ldots, m\}\} \cup (V_{\top,\bot} \setminus \{\bot_j\})\big)$.

We now turn to describe the transitions and actions. Player 0 is the only player that controls the vertices $1, \ldots, m$. From such a vertex i, she can choose a vertex $\langle \ell, i \rangle$ such that ℓ appears in c_i. For $1 \leq j \leq n$ and $1 \leq i \leq n$, the players that control the vertex $\langle x_j, i \rangle$ are $\top_j$ and 0. Similarly, players $\bot_j$ and 0 control the vertex $\langle \neg x_j, i \rangle$. These players choose an action in $\{0, 1\}$. If the XOR of their actions is 0, the game proceeds to vertex $\top_j$ in the first case and $\bot_j$ in the second. If the XOR is 1, the game proceeds to vertex $i+1$ if $i < m$, and otherwise to vertex 1. Finally, the vertices in $V_{\top,\bot}$ have only self loops. Note that $\mathcal{G}$ is 2-concurrent. Indeed, in all vertices, the transition depends on the actions of at most two players, thus $\mathcal{G}$ is polynomial in φ.

We argue that an NE profile in $\mathcal{G}$ corresponds to a satisfying assignment to the variables in φ. Let P be an NE profile in $\mathcal{G}$. First, we claim that Player 0's objective must be satisfied in $outcome(P)$. Otherwise, she can deviate to a strategy that chooses the action in every state $\langle \ell, i \rangle$ that avoids the vertices in $V_{\top,\bot}$. This is possible since her move is unilateral, and the other players' strategies stay as they are in P. Next, for $1 \leq j \leq n$, it is not possible that the objectives of both players $\top_j$ and $\bot_j$ are not satisfied. Otherwise, since $outcome(P)$ avoids $V_{\top,\bot}$, there are indices $1 \leq i_1, i_2 \leq m$ such that both $\langle x_j, i_1 \rangle$ and $\langle \neg x_j, i_2 \rangle$ are visited infinitely often. Since Player 0's strategy is as in P, then when Player $\top_j$ gets the chance, she would force the game to get stuck in vertex $\top_j$, and similarly for Player $\bot_j$. Thus, the assignment that sets x_j to be true if Player $\bot_j$'s objective is not satisfied, and otherwise sets x_j to be false, is a satisfying assignment. In Appendix A.2 we formalize this intuition and show the other, easier, direction.

We claim that $\mathcal{G}$ has an NE iff φ is satisfiable. For the first direction, assume that P is an NE profile in $\mathcal{G}$. We claim that φ is satisfiable. We start by claiming that for every variable x_j, $outcome(P)$ never visits either $\langle x_j, k\rangle$ or $\langle \neg x_j, k\rangle$, for every $1 \leq k \leq m$. That is, the play is "consistent" w.r.t to the polarity of x_j. Indeed, assume by way of contradiction that both $\langle x_j, k\rangle$ and $\langle \neg x_j, k'\rangle$ are visited in $outcome(P)$. Since the accepting vertices of players $\top_j$ and $\bot_j$ are disjoint, then w.l.o.g $\top_j \notin W(P)$. Then, when visiting $\langle x_j, k\rangle$, Player $\top_j$ can deviate to vertex $\top_j$ and gain.

Consider the assignment f that sets, for $1 \leq j \leq n$, the variable x_j to be true if Player $\bot_j$'s objective is not satisfied, and otherwise sets x_j to false. By the above, f is a legal assignment to the variables. We claim that it is a satisfying assignment. Observe that Player 0's objective is satisfied in $outcome(P)$, since Player 0 can always deviate to keep the game out of $V_{\top,\bot}$. Thus, for every $1 \leq i \leq m$, vertex i is visited infinitely often as well as some vertex $\langle \ell, i\rangle$ in $outcome(P)$. If $\ell = x_j$, for some $j \in \{1, \ldots, n\}$, then Player $\top_j$'s objective must be satisfied as otherwise he would deviate to force the game from $\langle \ell, i\rangle$ to vertex $\top_j$. Similarly, if $\ell = \neg x_j$, then Player $\bot_j$'s objective is satisfied. Our definition of f implies that in both cases $f(\ell) = true$, thus the clause c_i has a satisfied literal, and we are done.

Assume that φ is satisfiable, and let f be a satisfying assignment. We construct an NE profile P in $\mathcal{G}$. The strategies are as follows: for every $1 \leq j \leq n$, whenever players $\top_j$ and $\bot_j$ get a chance to move, they choose the action 0. For $1 \leq i \leq m$ let ℓ be a literal in c_i such that $f(\ell) = true$. From vertex i, Player 0 chooses to move to vertex $\langle \ell, i\rangle$. When reaching a vertex $\langle \ell, i\rangle$, for some $1 \leq i \leq m$, Player 0 chooses the action 1. Thus, the XOR of the two players actions in the vertex is 1, and the game proceeds to vertex $i + 1$ if $i < m$, and to vertex 1 otherwise.

We claim that P is an NE. First, Player 0's objective is satisfied as $outcome(P)$ does not reach a vertex in $V_{\top,\bot}$. Assume towards contradiction that P is not an NE, thus w.l.o.g there is $1 \leq j \leq n$ such that Player $\top_j$'s objective is not satisfied and she can benefit from deviating. Recall that Player $\top_j$ moves only in vertices of the form $\langle x_j, i\rangle$, for $1 \leq i \leq m$. Thus, there is such a vertex that is visited in $outcome(P)$. Thus, by the definition of Player 0's strategy in P, we have $f(x_j) = true$. Moreover, no vertex of the form $\langle \neg x_j, i\rangle$ is visited in $outcome(P)$, for $1 \leq i \leq m$. Since the vertices in $V_{\top,\bot}$ are also not visited in $outcome(P)$, Player $\top_j$'s objective is satisfied, thus we reach a contradiction, and we are done.

A.3 Correctness proof of the uniform-costs reduction of Theorem 4

For the first direction, assume there is a set cover $C \subseteq S$ of U with $|C| \leq \ell$. Consider the assignment f in which $f(v_{\text{end}}, S_j) = \top$ for every $S_j \in C$, and the rest of f is chosen to have cost 0. Thus, f repairs v_{end} to be accepting for the players S_j in C. Clearly, $cost(f) = |C| \leq \ell$.

We claim that $\mathcal{G}^f$ has an NE. Consider the profile P in which, for every $i \in U$, Player i moves from vertex i to vertex $\langle S_j, i\rangle$ such that $S_j \in C$ and $i \in S_j$, and from every vertex $\langle S_j, i\rangle$, the players $U \cup \{S_j\}$ cooperate and proceed to vertex $i + 1$ (or to v_{end}). We claim that this profile is an NE. Indeed, the outcome of P reaches v_{end} while only traversing vertices that are controlled by players for whom v_{end} is accepting in $\mathcal{G}^f$, namely players in U and C. Hence, none of these players has an incentive to deviate. The other players have no effect on the outcome of the profile, and thus cannot deviate to win. We conclude that $\mathcal{G}^f$ has an NE.

For the second direction, assume that f is an assignment of cost at most ℓ such that $\mathcal{G}^f$ has an NE, attained by some profile P. We first claim that $outcome(P)$ reaches v_{end}. Assume by way of contradiction that $outcome(P)$ does not reach v_{end}. We prove that then, there exists $u \in U$ such that $f(v_{\text{end}}, u) = \top$ and Player u's objective is not satisfied in $outcome(P)$. To prove the latter, we assume, again by way of contradiction, that this is false. Thus, for every $u \in U$, either $f(v_{\text{end}}, u) = \bot$ or Player u's objective is satisfied in $outcome(P)$, thus there is at least one vertex $v \neq v_{\text{end}}$ with $f(v, u) = \top$. Since $cost(v_{\text{end}}, u, \bot) = 1$ and $cost(v, u, \top) = 1$ for all $v \neq v_{\text{end}}$, the latter implies that f has cost at least n, which contradicts the fact that $cost(f) < n$. So, let $u \in U$ be such that

$f(v_{\text{end}}, u) = \top$ and Player u's objective is not satisfied in $outcome(P)$. The way we defined the transitions from vertices of the form $S \times U$ with a XOR implies that every player in U can direct the game by unilaterally deviating from her strategy in P from every vertex $\langle S_j, i\rangle$ to $i+1$ and eventually to v_{end}. Therefore, Player k can deviate from her strategy in P and force the game to reach v_{end}, in which case her objective is satisfied. Thus, the deviation is beneficial, contradicting the fact that P is an NE. This concludes the proof that $outcome(P)$ reaches v_{end}.

Since $outcome(P)$ reaches v_{end}, then setting $f(\langle S_j, i\rangle, S_j) = \bot$ has no effect on the set of players whose objectives are met in $outcome(P)$. Moreover, it is clearly not helpful for attaining an NE to set $f(v_{\text{end}}, i) = \bot$ for $i \in U$, as it only adds constraints on the profile being an NE. Thus, the only possible assignments are $f(v_{\text{end}}, S_j) = \top$ for some $S_j \in S$.

We claim that $C = \{S_j \in S : f(v_{\text{end}}, S_j) = \top\}$ is a set cover of U of size at most ℓ. First, since $cost(f) \leq \ell$, then $|C| \leq \ell$. Assume by way of contradiction that C is not a set cover of U. Then, there exists $i \in U$ that is not covered by C. Thus, at vertex i, Player i proceeds to a vertex $\langle S_j, i\rangle$ such that v_{end} is not winning for Player S_j. Again, the way we have defined the XOR transitions implies that Player S_j can unilaterally deviate from her strategy and force the game to stay in $\langle S_j, i\rangle$, for which $\langle S_j, i\rangle \in \alpha^f_{S_j}$. Hence, there cannot be an NE in $\mathcal{G}^f$, which is a contradiction. Thus, C is indeed a set cover, and we are done.

A.4 Correctness proof of the negative one-way costs reduction of Theorem 4

We now proceed to prove the correctness of the reduction. We claim that there exists an assignment f with $cost(f) \leq \ell$ such that $\mathcal{G}^f$ has an NE iff there is a set cover C of size at most ℓ. For the first direction, consider a set cover C of size at most ℓ. W.l.o.g $S = \{S_1, ..., S_\ell\}$. Let f be the assignment obtained by setting $f(S_j, 2) = \bot$ for $1 \leq j \leq \ell$, and by setting the rest of the values so that $f(v,t) \in free_{cost}(v,t)$ for $t \in \{1, 2\}$ and $v \in V$. Recall that $|free_{cost}(v,t)| = 1$, so f as above is unique. Clearly $cost(f) = |S| \leq \ell$. We claim that $\mathcal{G}^f$ has an NE. Indeed, player 1 can force Player 2 to lose if the game reaches s_1, since if player 2 chooses i, Player 1 would choose $S_j \in C$ such that $i \in S_j$, which is now rejecting for Player 2. Thus, Player 2 is doomed to lose, and has no incentive to deviate from the profile that goes to s_1, then cycles in v_1 and wins for Player 1. Conversely, assume there exists an assignment f such that $\mathcal{G}^f$ has an NE. Since $cost(f) \leq \ell$, then there exists a vertex $v \in \{v_1, ..., v_{\ell+1}\}$ such that $f(v, 1) = \top$. Thus, player 1 can force the game to reach and then stay forever in v. Accordingly, an NE is possible only when player 2 has no incentive to deviate, which means there exists a set $C \subseteq \{S_1, ..., S_m\}$ of size at most ℓ such that $f(S_j, 2) = \bot$ for every $S_j \in C$, and such that player 1 can choose, for every $i \in U$, a set $S_j \in C$ with $i \in S_j$. Thus, C is a set-cover of size at most ℓ.

A.5 Correctness proof of the algorithm in Theorem 5

We start by explaining in detail the intuition behind our algorithm.

The definitions in Appendix A.1 consider a given concurrent game. Here we consider partial games. Whenever we consider the game against i, we refer to the concrete two-player games obtained from $\mathcal{G}$ with the assignment that assigns $\bot$ to the vertices in DC_i. Thus, the coalition has a strategy that forces the game to visit F_i only finitely often from every vertex in C_i, and a transition is doomed for Player i if Player i cannot alter her action in $\overline{a}$ and escape C_i. Our algorithm checks whether there is a path τ to S that traverses only transitions in doomed(L_S). If so, it concludes that $\mathcal{G}$ can be repaired to have an NE with the assignment f that is defined as follows. For every $v \in V$, for $j \in L_S$, we have $f(v,j) = \bot$ and, for $j \in W_S$, we have $f(v,j) = \top$. Indeed, the profile whose outcome is τ followed by a path that visits all the vertices in S infinitely often, and which punishes a player that deviates from her expected action in τ is an NE in $\mathcal{G}^f$.

When the algorithm examines an ergodic SCC S and finds that it cannot serve as a witness, it removes S from $\mathcal{G}$ along with transitions that guarantee the soundness of such a removal. That is, in

function DON'T-CARE-BÜCHI($\mathcal{G}$)
  Let U be the set of transitions in $\mathcal{G}$.
  while $U \neq \emptyset$ **do**
    Let S be an ergodic SCC in $\mathcal{G}|_U$.
    if there is a path in $\mathcal{G}$ from v_0 to S that uses only edges from doomed(L_S) **then**
      return YES
    $U \leftarrow U \setminus \Delta(S)$
    while not fixed point **do**
      Let T be an ergodic SCC in $\mathcal{G}|_U$.
      Remove from U every edge in $\Delta(T) \setminus$ doomed(L_T).
    end while
    Remove every transition $\langle v, \overline{a}, v' \rangle \in U$ if no infinite path starts in v'.
  end while
  return NO

Figure 5 Solving the SR problem for Büchi games.

addition to the removal of S, the algorithm removes transitions so that if an ergodic SCC T is a witness for the existence of NE in the updated game $\mathcal{G}'$, then there is a repair so that $\mathcal{G}$ has an NE. The algorithm removes transitions so that every ergodic SCC T in $\mathcal{G}'$ consists only of transitions in doomed(L_T). Consider a witness ergodic SCC T in $\mathcal{G}'$ with the assignment f as described in the above. Let P be the corresponding NE profile such that $outcome(P)$ gets stuck in T. We claim that P is an NE in $\mathcal{G}^f$. Indeed, since the transitions in T are doomed for all the players in L_T, no losing player can deviate so that her objective is satisfied. Accordingly, the algorithm can apply to T with respect to $\mathcal{G}'$ the same steps applied to an ergodic SCC in $\mathcal{G}$.

Note that removing transitions that are not doomed from an ergodic SCC might cause it to be disconnected. Thus, the process of removing transitions is iterative. The algorithm finds an eroding SCC T and removes the transitions that are not doomed for L_T. Then, it finds another ergodic SCC (possibly an SCC that is contained in T) and performs the same removals. The process continues until a fixed point is reached. Finally, the algorithm removes transitions that do not participate in an infinite path. At this point the game has the property we require in the above.

Before we describe the algorithm we introduce some notation. Let $\Delta \subseteq V \times A_1 \times \cdots \times A_k \times V$ be the transitions of $\mathcal{G}$. Thus, $\langle v, \overline{a}, v' \rangle \in \Delta$ iff $\delta(v, \overline{a}) = v'$. For a set of vertices $S \subseteq V$, let $\Delta(S)$ be the transitions that include vertices in S, thus $\Delta(S) = \Delta \cap S \times A_1 \times ... \times A_k \times S$.

We are now ready to describe the algorithm, which appears in Figure 5.

The algorithm terminates after at most $|\Delta|$ iterations of the outer `while` loop as at least one vertex is removed in every iteration of this loop. Finding the sets C_i, for every $i \in \Omega$ can can also be done in polynomial time as it requires constructing and solving $|\Omega|$ concurrent two-player Büchi games, as explained in Section 2.3. Once the sets are found, deciding whether a transition $\langle v, \overline{a}, v' \rangle$ is doomed for $i \in \Omega$ boils down to checking, for every $a'_i \in A_i$, whether $\delta(v, \overline{a}[i \leftarrow a'_i])$ is in C_i.

We prove that the algorithm returns "yes" iff there is an assignment f such that $\mathcal{G}^f$ has an NE. For the first direction, assume the algorithm returns "yes". Let S be an ergodic SCC in $\mathcal{G}|_U$ and τ the *witness* path that leads to S. Consider the assignment f that is defined as follows. For every $v \in V$, for $j \in L_S$, if $\bot \in free_{cost}(v, j)$, we define $f(v, j) = \bot$ and, for $j \in W_S$, if $\top \in free_{cost}(v, j)$, we define $f(v, j) = \top$. Moreover, consider a profile P in which the players play a strategy which results in an outcome that traverses τ and repeats indefinitely some cycle τ' that visits all vertices in S. Note that the objective of every player in W_S is satisfied in this outcome, so none of these players have an incentive to deviate. Next, we set the strategies in P so that a unilateral deviation is not beneficial for any player in L_S. We can prove by induction that $\Delta(S)$ consists only of transitions in doomed(L_S). Thus, both τ and τ' traverse only transitions in doomed(L_S). Let $i \in L_S$ and a vertex v in τ or τ', and assume the expected vector of action in v is $\overline{a} \in A$. Assume Player i performs an action $a'_i \in A_i$ different from the one he is expected to perform in $\overline{a}$. Since τ and τ' use only transitions in doomed(L_S), we have $\delta(v, \overline{a}[i \leftarrow a'_i]) = v' \in C_i$. We set the players' strategies in P so that the players $\Omega \setminus \{i\}$ (the coalition) switch to their winning strategy in the Büchi game against i from v'. Thus, the outcome of the game is a play that visits F_i only finitely often, and Player i has no incentive to deviate. Clearly, P is an NE in $\mathcal{G}^f$.

For the second direction, assume towards contradiction that the algorithm returns "no" and there is an assignment f and profile P in $\mathcal{G}^f$ that is an NE. Let $\tau = outcome(P)$. Let $W(\tau) \subseteq \Omega$ be the set of players whose objective is satisfied in P and $L(\tau) = \Omega \setminus W(\tau)$ be the set of players whose objective is not satisfied in τ. Let $inf(\tau)$ and $\Delta(\tau)$ be the vertices and transitions that are visited by τ infinitely often, respectively. Consider an SCC S with $inf(\tau) \subseteq S$. Note that $W(\tau) \subseteq W_S$ and $L_S \subseteq L(\tau)$. For $i \in L(\tau)$, let C_i^f be the set of winning vertices for the coalition in the Büchi game between Player i and $\Omega \setminus \{i\}$ with objective $F_i \cup \{v : f(v,i) = \top\}$. Since the Büchi objective is a superset of F_i, we have $C_i^f \subseteq C_i$. Finally, similarly to the above, we define doomed$(f, L(\tau))$ to be the transitions in which players in $L(\tau)$ cannot escape the vertices C_i^f. Thus, we have $\bigcap_{i \in L(\tau)} C_i^f \subseteq \bigcap_{i \in L_S} C_i$ and doomed$(f, L(\tau)) \subseteq$ doomed(L_S). Clearly, τ traverses only edges in doomed$(f, L(\tau))$ otherwise a player in $L(\tau)$ can benefit from a unilateral deviation.

Consider the first time the algorithm removes a transition in $\Delta(\tau)$ from U. We distinguish between three cases. In the first case, this occurs in Line 7 as part of a removal of a complete ergodic SCC S. Since $inf(\tau)$ is strongly connected, we have $inf(\tau) \subseteq S$. Assume w.l.o.g that the vertices in $\tau \setminus inf(\tau)$ form an acyclic path τ from v_0 to $inf(\tau)$. By the above, τ uses only transitions in doomed(L_S), thus we reach a contradiction to the fact that the algorithm does not terminate with "yes". For the second case, the first removal of a transition in $\Delta(\tau)$ occurs in Line 10. Thus, there is an ergodic SCC T in $\mathcal{G}|_U$ with $t \in \Delta(S)$ and $t \in \overline{C_i}$ for some $i \in L_T$. Again, since $inf(\tau)$ is strongly connected, we have $inf(\tau) \subseteq T$. Recall that $\overline{C_i} \subseteq \overline{C_i^f}$ and $L(\tau) \subseteq L_S$. Thus, Player i's objective is not satisfied in τ, and she has a strategy ρ such that $outcome(v, P[i \leftarrow \rho])$ visits her objective infinitely often. In other words, Player i can benefit from a unilateral deviation when the game reaches t, and we reach a contradiction to the fact that P is an NE. Finally, the first removal cannot occur in Line 12 as $inf(\tau)$ is strongly connected and there is an infinite path that starts in any one of its vertices, and we are done.

A.6 Theorems and proofs of Section 3.2

▶ **Theorem 10.** *The problem of deciding whether a concrete reachability or co-Büchi game with a constant number of players has an NE can be solved in polynomial time, and it is in* NP $\cap$ coNP *for parity objectives.*

Proof. We remark that the case of reachability games was proved in [2]. We bring the proof here for completeness. Consider a game $\mathcal{G}$. As we showed in Section 2.4, the NE existence problem can be solved for reachability and co-Büchi games in NP, and the solution amounts to nondeterministically guessing a path in $\mathcal{G}$, and verifying that the transitions taken along the path are doomed for the losers in the path. When the number of players is constant t, we can proceed as follows. For every $W \subseteq \Omega$, consider the graph obtained from $\mathcal{G}$ by leaving only the transitions that correspond to transitions in doomed$(\Omega \setminus W)$ (which can be computed in polynomial time by solving $|\Omega|$ two-player zero-sum safety or Büchi games). Then, for reachability objectives, check if there is a path that is winning for all the players in W. Such a path induces an NE profile. For co-Büchi objectives, check if there is a reachable SCC S such that $S \subseteq \bigcap_{i \in W} \alpha_i$. A path to S, which then traverses S indefinitely, induces an NE profile.

Since there are only 2^t subsets of Ω, this algorithm works in polynomial time.

For parity objectives, membership in NP follows from the fact that the problem for arbitrary number of players is in NP. Membership in coNP is similar to the above. For $i \in \Omega$, recall that the problem of solving the game against i is in coNP. Thus, the algorithm first finds the cage C_i, for every $i \in \Omega$. Then, for every $W \subseteq \Omega$, consider the graph obtained from $\mathcal{G}$ by leaving only the transitions that correspond to transitions in doomed$(\Omega \setminus W)$. The algorithm finds an SCC S that is reachable from v_0 for which $\max\{\alpha_i(s) : s \in S\}$ is even, for every $i \in W$. Again, since there are only 2^t subsets of Ω, the algorithm clearly runs in nondeteministic polynomial time, and shows that the problem is in coNP. ◀

▶ **Theorem 11.** *The SR problem for Büchi and reachability games with uniform or positive one-way costs and a constant number of players can be solved in polynomial time.*

Proof. Let $\mathcal{G}$ be a Büchi (resp. reachability) game with a constant number t of players, let $cost$ be a cost function that is either uniform or positive one-way, and let $k \in \mathbb{N}$ be a bound on the budget for the repair. If $k \geq t$, the budget enables us to please all players: in reachability games, by making v_0 – the initial vertex of $\mathcal{G}$, accepting for all players, and in Büchi games, by finding a self-reachable vertex and making it accepting for all players. Thus, in this case the algorithm always returns "yes".

If $k < t$, then k is a constant too. Consider a subset S of $\Omega \times V$. Each such subset induces an assignment f_S with cost $|S|$ that switches the assignment induced by $cost$ to all the pairs in S. For example, in the uniform case, f_S is such that for all $\langle i, v\rangle \in S$, if $cost(v, i, \top) = 0$ and $cost(v, i, \bot) = 1$, then $f_S(i, v) = \bot$, and dually for $\top$. Since there are polynomially many subsets S of size at most k and, by Theorem 10 and Section 2.4, checking for an NE in $\mathcal{G}^{f_S}$ can be done in polynomial time, and we are done. ◀

▶ **Theorem 12.** *The SR problem for reachability and co-Büchi games with don't-cares and a constant number of players can be solved in polynomial time, and the problem is in* $\mathrm{NP} \cap \mathrm{coNP}$ *for parity games.*

Proof. As we observed in the proof of Theorem 5, it is enough to consider assignments such that for every player i, all the "don't care" vertices are either set to $\top$ or $\bot$ (or the respective ranks in parity games). Since the number of players is constant, we can go over all such assignments and check the existence of an NE for each assignment in polynomial time. We now formalize this intuition.

Consider a reachability, co-Büchi, or parity game $\mathcal{G}$ with a don't-care cost function $cost$. For a set $X \subseteq \Omega$, we define the *natural assignment with respect to cost and* X as the assignment f_X that modifies all the don't cares to the value that makes satisfaction for these players' objectives easiest and modifies the don't cares for $i \in \Omega \setminus X$ so that satisfaction for these players' objectives hardest. Formally, for reachability and co-Büchi games, for all $i \in X$, we have that $f_X(i, v) = \top$ iff $\top \in \mathit{free}_{cost}(v, i)$, and for all $i \in \Omega \setminus X$, we have that $f_X(i, v) = \bot$ iff $\top \in \mathit{free}_{cost}(v, i)$. For parity games, for all $i \in X$, we have that $f_X(i, v)$ is the maximal even index in $\mathit{free}_{cost}(v, i)$, and for all $i \in \Omega \setminus X$, we have that $f_X(i, v)$ is the maximal odd index in $\mathit{free}_{cost}(v, i)$.

Recall that in don't-care costs all assignments f satisfy $cost(f) = 0$ or $cost(f) = \infty$. Consider an NE in $\mathcal{G}^f$ for some assignment f such that $cost(f) = 0$. Let $W \subseteq \Omega$ be the set of players who win in the outcome of the NE profile, and consider the natural assignment f_W. It is easy to see that the NE profile in $\mathcal{G}^f$ is also an NE in $\mathcal{G}^{f_W}$. Moreover, $cost(f) = cost(f_W) = 0$. Thus, in order to decide if there is an assignment with an NE with a winning set W, it is enough to consider natural assignments. Since there is a constant number t of players, there are only 2^t choices for W. Thus, in order to solve the SR problem with don't-cares, we proceed as follows. Given $\mathcal{G}$, for every $W \subseteq \Omega$, consider the natural assignment f_W, and check if there is an NE in $\mathcal{G}^{f_W}$. If one is found, return "yes". Otherwise, return "no". The correctness of the algorithm follows from the observation above. The complexity follows from the complexity of deciding the existence of an NE in games with a constant number of players. ◀

▶ **Theorem 13.** *The SR problem for Büchi, co-Büchi, and reachability games with negative one-way costs and a constant number of players is NP-complete.*

Proof. The upper bound follows from Theorems 3 and 4. For the lower bound, recall that the reduction in the proof of Theorem 4 actually generates a game with two players. Also, in that reduction, in every assignment of finite cost, the only possible accepting vertices have self loops. Thus, the reduction applies to Büchi, co-Büchi and reachability winning conditions, which are thus NP-hard also for a constant number of players.

◀

▶ **Theorem 14.** *The SR problem for co-Büchi and parity games and uniform costs with a constant number of players is NP-complete.*

Proof. Membership in NP follows Theorem 3.

For the lower bound we show that co-Büchi games are NP-hard by a reduction from SET-COVER. The reduction is similar to the one in the case of negative one-way costs in the proof of Theorem 4, except that we replace some of the vertices by cycles of length $\ell + 1$. In co-Büchi games, this has the effect that no assignment can make an entire cycle accepting (unless it has cost 0). Therefore, effectively, only negative one-way assignments are useful. We now describe the construction and proof in detail.

Consider an input $\langle U, S, \ell \rangle$ for SET-COVER, where $U = \{1, \ldots, n\}$, $S = \{S_1, \ldots, S_m\}$, and $\ell \in \mathbb{N}$. Assume w.l.o.g $\ell < \min\{n, m\}$. We construct a game $\mathcal{G}$ with 2 players as follows. The vertices are $U \cup (S \times \{1, ..., \ell + 1\}) \cup \{s_0, s_1, s_2\} \cup (\{v_1, ..., v_{\ell+1}\} \times \{1, ..., \ell + 1\})$. The game starts in s_0, where the actions for the players are $\{0, 1\}$. If the XOR of the actions is 0, the game moves to vertex s_1, where Player 1 chooses a vertex from $\langle v_1, 1\rangle, ..., \langle v_{\ell+1}, 1\rangle$, and from $\langle v_i, 1\rangle$ starts a cycle through $\langle v_i, 1\rangle, \langle v_i, 2\rangle, ..., \langle v_i, \ell+1\rangle, \langle v_i, 1\rangle$. We set $cost(\langle v_i, r\rangle, 1, \top) = 0$ for $1 \le i \le \ell+1$ and $1 \le r \le \ell+1$. If the XOR in s_0 was 1, the game proceeds to vertex s_2 from which player 2 chooses a vertex $i \in U$. In vertex i, player 1 chooses a vertex $\langle S_j, 1\rangle$ such that $i \in S_j$. Then, the game continues in a cycle through $\langle S_j, 2\rangle, ..., \langle S_j, \ell + 1\rangle, \langle S_j, 1\rangle$. We set $cost(\langle S_j, r\rangle, 2, \top) = 0$ for $1 \le j \le m$ and $1 \le r \le \ell + 1$. The rest of the $cost$ function is set to give $\perp$ cost 0, and is completed to be a uniform cost.

We claim that there exists an assignment f with $cost(f) \le \ell$ such that $\mathcal{G}^f$ has an NE iff there is a set cover C of size at most ℓ. For the first direction, consider a set cover C of size at most ℓ. W.l.o.g $C = \{S_1, ..., S_\ell\}$. Let f be the assignment obtained by setting $f(\langle S_j, 1\rangle, 2) = \perp$ for $S_j \in C$, and by setting the rest of the values according to *free*. Clearly $cost(f) = |C| \le \ell$. We claim that $\mathcal{G}^f$ has an NE. Indeed, player 1 can force Player 2 to lose if the game reaches s_1, since if player 2 chooses i, Player 1 will choose $\langle S_j, 1\rangle \in C$ such that $i \in S_j$, and the cycle through the $\langle S_j, r\rangle$ cycle does not satisfy the co-Büchi condition for Player 2. Thus, Player 2 has no incentive to deviate from the play that goes to $\langle v_1, 1\rangle$, then remains in the cycle and wins for Player 1.

Conversely, assume there exists an assignment f such that $\mathcal{G}^f$ has an NE. Since $cost(f) \le \ell$, then there exist $1 \le i \le \ell + 1$ such that for every $1 \le r \le \ell + 1$ we have $f(\langle v_i, r\rangle, 1) = \top$. Thus, in every profile, player 1 can deviate and force the game to end in $\langle v_i, 1\rangle$ and win from there. Accordingly, an NE is possible only when player 2 has no incentive to deviate. Since f cannot assign $\top$ to an entire cycle in $\langle v_i, 1\rangle, ..., \langle v_i, \ell + 1\rangle$, then player 2 cannot win in an NE. Thus, an NE is only possible when Player 2 loses. This means there exist a set $C \subseteq \{S_1, ..., S_m\}$ of size at most ℓ such that player 1 can choose, for every $i \in U$, a vertex $\langle S_j, 1\rangle$ which is set to $\perp$ for Player 2, with $i \in S_j \in C$, which means there is a set-cover of size at most ℓ.

◀

A.7 Proof of Theorem 6

The running time of the algorithm is polynomial in the size of $\mathcal{G}$ and exponential in $|\Omega|$, and since we assume $|\Omega|$ is constant, the running time of the algorithm is polynomial. We turn to prove its correctness. Assume the algorithm outputs "yes". Then, there is a set of players $W \subseteq \Omega$ for which there is a vertex $v \in V$, a path τ_1 from v_0 to v in G_W, and a cycle τ_2 in G_W from v to itself with weight at most p. We define an assignment f as follows. For every vertex u that τ_2 traverses and for every $i \in W$ we set $f(u, i) = \top$. The assignment in the rest of the vertices s and players j is set such that $f(s, j) \in free_{cost}(s, j)$ (and since $|free_{cost}(s, j)| = 1$, this assignment is unique).

By our definition of weights, and since the weight of τ_2 is at most p, we have $cost(f) \le p$. We claim that $\mathcal{G}^f$ has an NE. Indeed, consider the profile P in which the players cooperate so that $outcome(P) =$

$\tau_1 \cdot \tau_2^\omega$. Since G_W includes only transitions that are doomed for L, when Player i, for $i \in L$, performs an action that is not expected of her, the game remains in the vertices C_i. We set the players' strategies so that when such a violation occurs, they change their strategy to one that keeps Player i from satisfying her goal. Clearly, P is an NE.

For the other direction, assume there is an assignment f and an NE P in $\mathcal{G}^f$. Let $\tau = outcome(P)$. Note that τ must use only transitions that are doomed for $L(P)$ in $\mathcal{G}^f$, as otherwise P is not an NE. Thus, τ is a path in $G_{W(P)}$. Let $v \in inf(\tau)$. Consider a simple cycle ρ that includes v and traverses only vertices in $inf(\tau)$. Since $cost(f) \leq p$, we have that f performs at most p modifications to the vertices in ρ. Then, the algorithm will terminate with "yes" when reaching $W(P)$, v, and ρ, and we are done.

A.8 Proof of Theorem 7

For the lower bounds, it is easy to see that the SR problem can be reduced to the RSR problem, say by setting all rewards and q to the same value. For the upper bounds, it is not hard to see that whenever the SR problem is in NP, so is the RSR problem. Indeed, one can start by nondeterministically guessing a set $W \subseteq \Omega$ of winning players, verify that $\zeta(W) \geq q$, and then proceed to nondeterministically solve the SR problem while requiring the winning set of players to contain W (recall that ζ is monotone, so containing W is sufficient).

Thus, the only cases of interest are when the SR problem can be solved in polynomial time.

We start with Büchi games with don't cares, in the setting of an arbitrary number of players. Consider the algorithm in the proof of Theorem 5. We modify the algorithm to work for the RSR problem to check, whether an ergodic SCC S witnesses an NE, also whether $\zeta(W_S) \geq q$. if so, then the algorithm returns "yes". Otherwise, it continues.

We claim that the new algorithm solves the RSR problem. Clearly, by the correctness of the algorithm there, if the algorithm returns "yes" then there is an NE P with $\zeta(W(P)) \geq q$. Conversely, assume that there exists an NE P with $\zeta(W(P)) \geq q$, but that the algorithm returns "no". Carefully following the proof of Theorem 5 shows that in this case, the first removal of an edge in $S = inf(outcome(P))$ is done while considering an SCC S' that witnesses an NE for which $S \subseteq S'$, and therefore $W_S \subseteq W_{S'}$. Then, $\zeta(W_{S'}) \geq \zeta(W_S) \geq q$, so the algorithm would have returned "yes".

We proceed to the algorithms that handle games with a constant number of players. We observe that apart from the case of Büchi games with don't care costs, which was dealt above, our algorithms in Theorems 11 and 6 work by first fixing a set of winning players W, and then looking for a solution for the SR problem where the NE profile P has $W(P) \subseteq W$. Thus, for the RSR problem, it is enough to first check if $\zeta(W) \geq q$, and continue only if this is the case.

A.9 Transition repair

We start by formally defining the model of transition repair. We are given a game $\mathcal{G} = \langle \Omega, V, A, v_0, \delta, \{\alpha_i\}_{i\in\Omega}\rangle$ and a redirection cost function $cost : Q \times A_1 \times \ldots \times A_k \times Q \to \mathbb{N}_\infty$, which describes, for every $q, q' \in V$ and $\overline{a} \in A_1 \times \cdots \times A_k$, the cost of redefining $\delta(q, \overline{a})$ to be q', We require that for all $q \in V$ and $\overline{a} \in A_1 \times \cdots \times A_k$, we have $cost(q, \overline{a}, \delta(q, \overline{a})) = 0$.

A repair in this setting is simply a new transition function $\mu : Q \times A_1 \times \cdots \times A_k \to Q$, describing the new transitions. The repaired game is $\mathcal{G}^\mu = \langle \Omega, V, A, v_0, \mu, \{\alpha_i\}_{i\in\Omega}\rangle$, and the cost of the repair is the sum of the costs of the modifications from δ to μ. Formally, $cost(\mathcal{G}^\mu) = \sum_{q,\overline{a}\in Q\times A_1\times\cdots\times A_k} cost(q, \overline{a}, \mu(q, \overline{a}))$.

The *transition repair* problem (TR, for short) is defined as follows. Given a game $\mathcal{G}$, a cost function $cost$, and a threshold p, the TR problem is to decide whether there exists a repair μ such that $cost(\mathcal{G}^\mu) \leq p$ and the game $\mathcal{G}^\mu$ has an NE.

We split the proof of Theorem 8 to several theorems. We start by studying the general case.

▶ **Theorem 15.** *The TR problem is NP-complete.*

Proof. We start with membership in NP. Given a game $\mathcal{G}$ and a threshold p, we guess a repair μ of cost at most p. We then continue with the nondeterministic algorithm for deciding the existence of NE in $\mathcal{G}^\mu$. For the lower bound, the TR problem is clearly harder than deciding the existence of NE, thus we are left to show that the TR problem is NP-hard for Büchi objectives. We use a reduction similar to the one we used in Theorem 4 to show that the SR problem with negative one-way costs is NP-hard. ◀

We study the setting with a constant number of players. We note that the reduction in Theorem 15 uses two players and can be applied to all types of objectives, thus the lower bound applies also to this case. Clearly, the upper bound of Theorem 15 is also valid. Thus, we have the following.

▶ **Theorem 16.** *The TR problem with a constant number of players is NP-complete.*

In the SR problem, we studied special cost functions. It is not hard to prove that the NP-hardness proof applies also to the restricted cost function in which all costs of modified transitions are either 1 of ∞. The only case where we can go down to a polynomial algorithm is when all modifications are possible, and their cost is uniform. That is, for every $q, q' \in Q$ and $\overline{a} \in A_1 \times \ldots \times A_k$, if $\delta(q, \overline{a}) = q'$, then $cost(q, \overline{a}, q') = 0$, and otherwise $cost(q, \overline{a}, q') = 1$. Thus, interestingly, increasing the freedom of the repairs (the problem is NP-hard when $cost(q, \overline{a}, q') \in \{1, \infty\}$ for $q' \neq \delta(q, \overline{a})$) reduces the complexity of the TR problem. Formally, we have the following.

▶ **Theorem 17.** *The complexity of the TR problem with a uniform cost function is as follows. For an arbitrary number of players, it is NP-complete for all winning conditions. For a constant number of players, it can be solved in polynomial time for reachability and Büchi objectives, and is NP-complete for co-Büchi and parity objectives.*

Proof. We start with arbitrarily many players. The upper bound is as in Theorem 15. For the lower bound, we first observe that by setting p to 0, the TR problem coincide with the problem of NE existence, thus NP-hardness for all objectives but Büchi follow from the NP-hardness of the latter. We prove that the problem is NP-hard also for Büchi objectives.

We show a reduction from SET-COVER. Consider an input $\langle U, S, \ell \rangle$ to SET-COVER, where recall that $U = \{1, \ldots, n\}$, $S = \{S_1, \ldots, S_m\}$, and ℓ is a threshold. We assume w.l.o.g that $\ell \leq \min\{n, m\}$. We construct a game $\mathcal{G} = \langle U, \{v_0\} \cup U \cup S, A, v_0, \delta, \{\alpha_i\}_{i \in \Omega} \rangle$, where $\alpha_i = \{i\} \cup \{S_j \in S : i \in S_j\}$, and we describe A and δ in the following.

Intuitively, there are $\ell + 1$ parallel transitions from v_0 to every vertex $i \in U$, and $\ell + 1$ self loops on every vertex $i \in U$. Thus, no matter what transition repair we apply to the game, assuming that players $U \setminus \{i\}$ fix a strategy, Player i can force the game to proceed from v_0 to i and stay there indefinitely. Hence, in an NE profile in a repaired game, the objectives of all players are satisfied. The vertices S are not reachable from the initial vertex but have a transition "back" to v_0. A repair that corresponds to a set cover $S' \subseteq S$ moves $|S'|$ self loops of v_0 to point to the vertices in S'. Then, an outcome of an NE profile traverses the vertices S' by going back and forth between them and v_0. Clearly, the reduction is polynomial in the size of the input.

We formalize the construction by describing A and δ. For every $i \in \Omega$, the actions of Player i are $A_i = \{0, 1\}^x$, where $x = \lceil \log(\ell + n(\ell + 1)) \rceil$. There are 2^x outgoing transitions from every vertex. We order the transitions arbitrarily. Then, given actions $\overline{a} = a^1, \ldots, a^n \in A_1 \times \ldots \times A_n$, which are vectors over $\{0, 1\}^x$, we construct a vector $\tilde{a} \in \{0, 1\}^x$ by taking the XOR in each coordinate. Thus, for all $1 \leq i \leq x$, we have $\tilde{a}_i = \text{XOR } a_i^j$. Note that $\tilde{a}$ can be thought of as a number between 1 and 2^x. For every vertex $v \in V$, we define $\delta(v, \overline{a})$ to be the $\tilde{a}$-th outgoing transition from v. Note that for every $i \in U$ and $v \in V$, assuming that players $U \setminus \{i\}$ fix an action, Player i can force any of the 2^x outgoing transitions from v to be selected.

We continue to describe δ. For $i \in U$, there are $\ell + 1$ transitions from v_0 to i. The remaining $n(\ell + 1) - 1 \geq \ell$ transitions from v_0 are self-loops. For every $i \in U$, all the outgoing transitions are self loops, and for $S_j \in S$, all outgoing transitions are self loops apart from one transition to v_0.

We prove the correctness of the reduction. Assume there is a set cover $S' \subseteq S$ with $|S'| \leq \ell$. We construct a transition repair by choosing $|S'|$ self loops of v_0 and redirecting each one to a vertex in S'. Consider the profile P in the repaired game whose outcome traverses all the vertices in S' infinitely often. Since S' is a set cover, the objectives of all the players are satisfied, and P is clearly an NE.

For the second direction, assume there is a transition-repair μ and an NE profile P in $\mathcal{G}^\mu$. As in the above, all players are satisfied in P. Consider a vertex $i \in U$ that is visited in $outcome(P)$. Since all players are satisfied in P and i belongs only to α_i, we have that $outcome(P)$ cannot get stuck in i. Since i has only self loops in $\mathcal{G}$, the repair had to move one of its loops. So, the repair "pays" 1 for every state in U that is visited in $outcome(P)$. Clearly, the repair pays 1 for every state $S_j \in S$ that is visited in $outcome(P)$ as these states are not reachable in $\mathcal{G}$. Consider the set $S' \subseteq S$ that includes every $S_j \in S$ that is visited by $outcome(P)$, and, for every vertex $i \in U$ that is visited in $outcome(P)$, there is an arbitrary set $S_j \in S'$ such that $i \in S_j$. Since the repair costs at most ℓ, we have $|S'| \leq \ell$. Since all players are satisfied in P, S' is a set cover, and we are done.

We now proceed to study the case of a constant number of players. We prove the upper bound for Büchi games. The proof for reachability games is similar. Consider a Büchi game $\mathcal{G} = \langle \Omega, V, A, v_0, \delta, \{\alpha_i\}_{i \in \Omega} \rangle$. We show that there is always a repair of size $|\Omega|$ that guarantees a profile in which all the players' objectives are satisfied. We assume that for every $i \in \Omega$, there is a vertex $v_i \in \alpha_i$ such that v_i is reachable from the initial vertex, thus there is a pathπ_i from v_0 to v_i. Moreover, we assume that every such vertex has at least one outgoing transition. We repair the game so that there is a path from every v_i back to v_0, thereby closing a cycle. Then, a profile whose outcome traverses each of these loops infinitely often is an NE as all the players' objectives are satisfied. A first attempt to achieve this would be to modify an outgoing transition from every v_i so that its destination is v_0. However, if, for $j \neq i$, the path π_j uses the modified transition, then, v_j might be disconnected from v_0. So, we construct the repair more carefully. Let $\Pi \subseteq \{\pi_i : \pi_i \text{ is not a strict prefix of } \pi_j, \text{ for } j \neq i\}$. Consider the repair μ that redirects an outgoing transition from the last vertex v_i in π_i to v_0, for every $\pi_i \in \Pi$. As in the above, the profile that traverses each one of the loops $\pi_i v_0$ infinitely often, is an NE in $\mathcal{G}^\mu$. Clearly, $cost(\mathcal{G}^\mu) \leq |\Omega|$, and we are done.

We return to proving that the TR problem can be solved in polynomial time. Consider a game $\mathcal{G}$ and a threshold p. We distinguish between two cases. In the first case, we have $p \geq |\Omega|$, and the claim above implies that there is a repair that guarantees an NE. If $p \leq |\Omega|$, then it is constant, and we can go over all the possible repairs that alter at most p transitions and check whether one of them guarantees an NE.

The NP-hardness lower bound for the other objectives can be shown by using the reduction in Theorem 4 for negative one-way costs and adding parallel transitions. The upper bound follows from the case of arbitrary many players. ◀

A.10 Controlled players

We split the proof of Theorem 9 to several theorems.

▶ **Theorem 18.** *The CR problem for reachability, co-Büchi, and parity objectives is NP-complete.*

Proof. We start with membership in NP. Given a game $\mathcal{G}$ and a threshold p, we guess a subset of players, set their objectives to be the most permissive, and continue with the nondeterministic algorithm for deciding the existence of NE in the resulting game. For the lower bound, the CR problem is clearly harder than deciding the existence of NE (by setting $p = 0$). ◀

The interesting case is Büchi objectives, where the existence of an NE can be checked in polynomial time.

▶ **Theorem 19.** *The CR problem is in P for Büchi games.*

Proof. Consider a Büchi game $\mathcal{G} = \langle \Omega, V, A, v_0, \delta, \{\alpha_i\}_{i \in \Omega} \rangle$, a control cost function $cost : \Omega \to \mathbb{N}_\infty$, and $p \in \mathbb{N}$. Since controlling Player i amounts to setting $\alpha_i = V$, the CR problem amounts to deciding whether there exists $S \subseteq \Omega$ of cost at most p such that setting $\alpha_i = V$ for every $i \in S$ results in a game with an NE.

In general Büchi games, the description of δ is exponential in Ω. Thus, we can simply try every subset $S \subseteq \Omega$ of cost at most p. Since checking the existence of an NE in a Büchi game can be done in polynomial time, we are done. ◀

The proof of Theorem 19 is based on the size of the representation of the game. This raises the question of what happens when the game has a succinct representation, for example in c-concurrent games (see Remark 2.3).

▶ **Theorem 20.** *The CR problem is NP-complete for c-concurrent Büchi games.*

Proof. Membership in NP is easy, as given a game $\mathcal{G}$ and a threshold p we can guess a set of players S of cost at most p, and verify that the game has an NE when we control S.

For the lower bound, the proof is based on the same reduction from SET-COVER used in the proof of the positive one-way case of Theorem 4, after the modification explained in Remark 3.1.

Observe that in the reduction of Theorem 4 for the positive one-way case, every player in U can always force the game to reach v_{end}. Assuming $k < n, m$, it is not possible to ensure an NE by controlling k (or less) of the U players (since then some S_j player will deviate and force the game to not to reach v_{end}). Thus, the only way to ensure an NE is by setting v_{end} to be accepting for k S_j-players, which is similar to the repair done with positive one-way costs.

◀

Finally, we study the setting with a constant number of players. Similarly to the proof of the upper bound of Theorem 19, we can go over every subset of players, update their objectives to be the most permissive, and check if the resulting game has an NE. Since there are only a constant number of such sets, and since deciding the existence of NE with a constant number of players is in P, we have the following.

▶ **Theorem 21.** *For a constant number of players, the CR problem is in P for all objectives.*