Coverage Games

Orna Kupferman
School of Computer Science and Engineering, The Hebrew University, Jerusalem, Israel

Noam Shenwald
School of Computer Science and Engineering, The Hebrew University, Jerusalem, Israel

Abstract

We introduce and study *coverage games* – a novel framework for *multi-agent planning* in settings in which a system operates several agents but do not have full control on them, or interacts with an environment that consists of several agents.

The game is played between a *coverer*, who has a set of objectives, and a *disruptor*. The coverer operates several agents that interact with the adversarial disruptor. The coverer wins if every objective is satisfied by at least one agent. Otherwise, the disruptor wins.

Coverage games thus extend traditional two-player games with multiple objectives by allowing a (possibly dynamic) decomposition of the objectives among the different agents. They have many applications, both in settings where the system is the coverer (e.g., multi-robot surveillance, coverage in multi-threaded systems) and settings where it is the disruptor (e.g., prevention of resource exhaustion, ensuring non-congestion). We study the theoretical properties of coverage games, including determinacy, and the ability to a priori decompose the objectives among the agents. We solve the problems of deciding whether the coverer or the disruptor wins, analyze their tight complexity, and consider useful special cases.

2012 ACM Subject Classification Theory of computation → Formal languages and automata theory; Computing methodologies → Multi-agent planning

Keywords and phrases Two-Player Games, ω-Regular Objectives, Coverage, Planning

Digital Object Identifier 10.4230/LIPIcs.CONCUR.2025.27

Funding *Orna Kupferman*: This research is supported by the European Research Council, Advanced Grant ADVANSYNT.

1 Introduction

Synthesis is the automated construction of a system from its specification [23]. A *reactive* system interacts with its environment and has to satisfy its specification in all environments [15]. A useful way to approach synthesis of reactive systems is to consider the situation as a *two-player game* between the system and the environment [3]. The game is played on a graph whose vertices are partitioned between the players. Starting from an initial vertex, the players jointly move a token and generate a *play*, namely a path in the graph, with each player deciding the successor vertex when the token reaches a vertex she owns. The system wins if it has a strategy to ensure that no matter how the environment moves the token, the generated play satisfies an objective induced by the specification.

Reactive systems typically have to satisfy a conjunction of requirements, giving rise to games with *multiple objectives*. For example, in games with *generalized Büchi* objectives [6], the specification to the system induces a set $\{\alpha_1, \ldots, \alpha_m\}$ of subsets of vertices, and a play satisfies the specification if it visits infinitely often each of the sets α_l, for $1 \leq l \leq m$. Researchers have also considered *generalized parity* [7], *generalized reachability* [13], and *generalized reactivity* (GR(1)) [22, 6] objectives. In addition to the deterministic turn-based setting, games with multiple objectives have been studied in various richer settings, like concurrent, stochastic, and energy games [4, 8, 5, 29], and also allow *weighted* multiple objectives [19].

© Orna Kupferman and Noam Shenwald;
licensed under Creative Commons License CC-BY 4.0
36th International Conference on Concurrency Theory (CONCUR 2025).
Editors: Patricia Bouyer and Jaco van de Pol; Article No. 27; pp. 27:1–27:23
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

The need to satisfy multiple objectives is also at the heart of *planning* in multi-agent systems [25]. In a typical setting there, the agents constitute a fleet of robots, drones, or autonomous cars that needs to accomplish various tasks, such as exploring an unknown terrain or accommodating requests from users at different locations [10, 25, 9, 24, 26]. For example, in *adversarial patrolling* [20], a fleet of robots needs to detect penetrations into a guarded area; in the *offline-coverage* problem [1], the robots are required to visit every point in a work area, typically within the shortest possible time [8, 25, 9]; and in *pursuit evasion* [17], a robot aims to perform some task while avoiding a collision with an adversarial robot.

While the problems studied in planning in multi-agent systems involve adversarial factors such as physical limitations of the agents, bounded resources, or possible collisions, they assume full control over the agents. That is, they do not include an adversarial behavior of the agents themselves in the form studied in reactive synthesis.

We introduce and study *coverage games* – a framework for reasoning about planning tasks in which the system does not have full control over the agents. Formally, a coverage game is a two-player game with a set $\beta = \{\alpha_1, \ldots, \alpha_m\}$ of objectives, in which one player – the covering player (named *Coverer*) operates a number k of agents that together aim to cover all the objectives in β. In the beginning of the game, k tokens are placed on the initial vertex. Each agent is responsible for one token, and as in standard two-player games, the interaction of each agent with the second player – the disrupting player (named *Disruptor*) generates a play in the graph. The *outcome* of the game is then a set of k plays – one for each agent. The goal of Coverer is for these plays to cover all the objectives in β. That is, for all $1 \leq l \leq m$, there is an agent $1 \leq i \leq k$ such that α_l is satisfied in the play generated by the interaction of Agent i with Disruptor.

A *covering strategy* for Coverer is a vector of strategies, one for each agent, that ensures that no matter how Disruptor behaves, each objective is satisfied in at least one play in the outcome. In the *coverage problem*, we are given a game graph G, a set β of objectives, and a number $k \geq 1$ of agents, and we have to decide whether Coverer has a covering strategy.

Note that when $k = 1$, a coverage game coincides with a standard two-player game with multiple objectives. Also, it is not hard to see that when $k \geq m$, namely there are more agents than objectives, then a dominant strategy (in the game-theoretic sense) for Coverer allocates each objective to a different agent. The interesting cases are when $1 < k < m$, in which case the objectives need to be partitioned, possibly dynamically, among the agents.

Coverage games significantly extend the settings addressed in current studies of planning. For example, with Büchi underlying objectives, one can reason about *multi-robot surveillance*, where we need to ensure that each set of critical locations is visited infinitely often in at least one robot's patrol route. There, vertices owned by Disruptor correspond to locations in which we cannot control the movement of the robots, as well as positions in which the robots may encounter obstacles or experience a change in the landscape. Then, in *cyber-security systems*, the agents represent defense mechanisms, and the goal is to ensure that each set of potential attack vectors is mitigated infinitely often by at least one defense, countering an adversarial hacker. As another example, in *multi-threaded systems*, the agents are the processes, and we may want to ensure that each set of critical resources is accessed infinitely often, in all environments. Likewise, in *testing*, we want to activate and cover a set of functionalities of a software in all input sequences [11].

Unlike traditional two-player games, where the environment can be viewed as a system that aims to realize the negation of the specification, in coverage games Disruptor's objective is not to cover the negation of the objectives in β, but rather to prevent Coverer from covering all the objectives in β. Formally, a *disrupting strategy* for Disruptor is a strategy such that

for all strategies of Coverer, there is at least one objective in β that is not satisfied in each of the k generated plays. Note that while Coverer activates several agents, Disruptor follows a single strategy. This corresponds to environments that model the physical conditions of a landscape or servers whose response to users depends only on the interaction and not on the identity of the user. Then, in the *disruption problem*, we are given G, β, and k, and we have to decide whether Disruptor has a disrupting strategy. While two-player games are determined, in the sense that in all games, one of the two players has a winning strategy, we are going to see that coverage games need not be determined.

Viewing Disruptor as the system and the agents as operated by the environment further motivates the study of coverage games. For example, in *intrusion detection* systems, the agents of the environment are potential intruders, and the system must ensure that at least one set of critical access points remains blocked. In *cloud computing*, the environment consists of client processes competing for resources, and the system's goal is to prevent resource exhaustion. Finally, in *traffic management*, the environment agents are the vehicles, and the system ensures that at least one route remains uncongested. Note that now, Disruptor following a single strategy corresponds to systems like traffic controllers or vending machines, where the same policy is applied to all cars or customers that exhibit identical behavior.

A model related to coverage games from the point of view of the disruptor is *population games* [2]. These games focus on controlling a homogeneous population of agents that have the same non-deterministic behavior (see also [12, 14]). The edges of a two-player population game graph are labeled by actions. The game is nondeterministic, thus different edges that leave the same vertex may be labeled by the same action. As in the disruption problem, each agent moves a token along the graph. In each turn, a controller chooses the same action for all agents, and each agent decides how to resolve nondeterminism and proceeds to a successor vertex along an edge labeled with the action. The goal of the controller is to synchronize all the agents to reach a target state. The technical details of population games are very different from those of coverage games. Indeed, there, the controller takes the same action in all of its interactions with the agents, and the type of objectives are different from the coverage and disruption objectives in coverage games.

We study the coverage and disruption problems in coverage games with *Büchi* and *co-Büchi* objectives. We first study the theoretical aspects of coverage games. We warm up with games in which the number of agents enables an easy reduction to usual two-player games, and continue to *one-player coverage games*, where all vertices are owned by one player. In particular, when all vertices are owned by Coverer, the coverage problem boils down to finding k paths that cover all the objectives in β, and our complexity results are aligned with the NP-hardness of different variants of the *coverage path planning* problem for robot swarms [1].

We then continue to the general setting, show that it is undetermined, and examine the problem of an a-priori decomposition of the objectives in β among the agents. We show that the objectives cannot be allocated in advance, and characterize covering strategies for Coverer as ones in which all agents satisfy all objectives, as long as such decomposition is impossible. The characterization is the key to our upper bounds for the complexity of the coverage problem. We prove that the coverage problem is PSPACE-complete (Theorems 10 and 11), and that the disruption problem is Σ_2^{P}-complete (Theorems 15, 20, and 21), for both Büchi and co-Büchi objectives.

From a technical point of view, for the upper bounds, the main challenging results are the characterization of the decomposability of objectives and the recursive algorithm it entails (Theorems 9 and 10), as well as the ability to work with a short symbolic representation of strategies that are not memoryless (Lemma 19). For the lower bounds, the main technical issue is utilizing the agents to model satisfying assignments in Boolean satisfaction problems.

In Section 7 we discuss possible variants and extensions of coverage games. Beyond classical extensions of the underlying two-player games (e.g., concurrency, stochastic settings, partial visibility, etc.) and to the objectives (e.g., richer winning conditions, weighted objectives, etc.), as well as classical extensions from planning (optimality of agents and their resources), we focus on elements that have to do with the operation of several agents: communication among the agents (in our setting, the strategies of the agents are independent of each other), and the ability of Disruptor to also use different agents.

2 Preliminaries

2.1 Two-player games

A *two-player game graph* is a tuple $G = \langle V_1, V_2, v_0, E\rangle$, where V_1, V_2 are disjoint sets of vertices, owned by Player 1 and Player 2, respectively, and we let $V = V_1 \cup V_2$. Then, $v_0 \in V$ is an initial vertex and $E \subseteq V \times V$ is a total edge relation, thus for every $v \in V$, there is $u \in V$ such that $\langle v, u\rangle \in E$. The size of G, denoted $|G|$, is $|E|$, namely the number of edges in it. When we draw game graphs, the vertices in V_1 and V_2 are drawn as circles and squares, respectively.

In a beginning of a play in the game, a token is placed on v_0. Then, in each turn, the player that owns the vertex that hosts the token chooses a successor vertex and moves the token to it. Together, the players generate a *play* $\rho = v_0, v_1, \ldots$ in G, namely an infinite path that starts in v_0 and respects E: for all $i \geq 0$, we have that $\langle v_i, v_{i+1}\rangle \in E$.

A *strategy* for Player i is a function $f_i : V^* \cdot V_i \to V$ that maps prefixes of plays that end in a vertex owned by Player i to possible extensions of the play. That is, for every $\rho \in V^*$ and $v \in V_i$, we have that $\langle v, f_i(\rho \cdot v)\rangle \in E$. Intuitively, a strategy for Player i directs her how to move the token, and the direction may depend on the history of the game so far. A strategy is *memoryless* if its choices depend only on the current vertex and are independent of the history of the play. Accordingly, we describe a memoryless strategy for Player i by a function $f_i : V_i \to V$.

A *profile* is a tuple $\pi = \langle f_1, f_2\rangle$ of strategies, one for each player. The *outcome* of a profile $\pi = \langle f_1, f_2\rangle$ is the play obtained when the players follow their strategies in π. Formally, $\mathsf{outcome}(\pi) = v_0, v_1, \ldots \in V^\omega$ is such that for all $j \geq 0$, we have that $v_{j+1} = f_i(v_0, v_1, \ldots, v_j)$, where $i \in \{1, 2\}$ is such that $v_j \in V_i$.

A *two-player game* is a pair $\mathcal{G} = \langle G, \psi\rangle$, where $G = \langle V_1, V_2, v_0, E\rangle$ is a two-player game graph, and ψ is an objective for Player 1, specifying a subset of V^ω, namely the set of plays in which Player 1 wins. We discuss different types of objectives below. The game is zero-sum, thus Player 2 wins when the play does not satisfy ψ. A strategy f_1 is a *winning strategy* for Player 1 if for every strategy f_2 for Player 2, we have that Player 1 wins in $\langle f_1, f_2\rangle$, thus $\mathsf{outcome}(\langle f_1, f_2\rangle)$ satisfies ψ. Dually, a strategy f_2 for Player 2 is a winning strategy for Player 2 if for every strategy f_1 for Player 1, we have that Player 2 wins in $\langle f_1, f_2\rangle$. We say that Player i *wins* in $\mathcal{G}$ if she has a winning strategy. A game is *determined* if Player 1 or Player 2 wins it.

For a play $\rho = v_0, v_1, \ldots$, we denote by $\mathit{inf}(\rho)$ the set of vertices that are visited infinitely often along ρ. That is, $\mathit{inf}(\rho) = \{v \in V : \text{there are infinitely many } i \geq 0 \text{ such that } v_i = v\}$. For a set of vertices $\alpha \subseteq V$, a play ρ satisfies the *Büchi* objective α iff $\mathit{inf}(\rho) \cap \alpha \neq \emptyset$. The objective dual to Büchi is *co-Büchi*. Formally, a play ρ satisfies a co-Büchi objective α iff $\mathit{inf}(\rho) \cap \alpha = \emptyset$. We use $\gamma \in \{\mathrm{B}, \mathrm{C}\}$ to denote the different objective types.

For a play ρ and a set of objectives $\beta = \{\alpha_1, \ldots, \alpha_m\}$, we denote by $\mathsf{sat}(\rho, \beta)$ the set of objectives $\alpha_i \in \beta$ that are satisfied in ρ. An *All objective* is a set β of objectives, all of the same type. A play ρ satisfies an All objective β iff ρ satisfies every objective in β. That is,

if $\mathsf{sat}(\rho, \beta) = \beta$. The objective dual to All is *Exists*.[1] Formally, a play ρ satisfies an Exists objective β iff ρ satisfies at least one objective in β. That is, if $\mathsf{sat}(\rho, \beta) \cap \beta \neq \emptyset$. Note that an All-co-Büchi objective β is equal to the co-Büchi objective $\cup\beta$.

2.2 Two-player multi-agent coverage games

In coverage games, Player 1 operates a number of agents. All agents play against Player 2, and so the outcome of the game is a set of plays – one for each agent. Player 1 has multiple objectives, and her goal is to cover all the objectives, in the sense that each objective is satisfied in at least one of the plays in the outcome. Accordingly, we refer to Player 1 as the covering player, named *Coverer*, and refer to Player 2 as the disrupting player, named *Disruptor*.

Formally, for $\gamma \in \{\text{B,C}\}$, a *two-player multi-agent γ-coverage game* (γ-CG, for short) is a tuple $\mathcal{G} = \langle G, k, \beta \rangle$, where G is a two-player game graph, $k \in \mathbb{N}$ is the number of agents Coverer operates, and $\beta = \{\alpha_1, \ldots, \alpha_m\}$ is a set of objectives of type γ. When γ is not important or is clear from the context, we omit it and refer to $\mathcal{G}$ as a CG.

For $k \geq 1$, let $[k] = \{1, \ldots, k\}$. A *strategy* for Coverer is a tuple $F_1 = \langle f_1^1, \ldots, f_1^k \rangle$ of k strategies for Coverer in the game graph G. We assume that Disruptor uses the same strategy f_2 against all the agents of Coverer. A *profile* is a tuple $\pi = \langle F_1, f_2 \rangle$ of strategies $F_1 = \langle f_1^1, \ldots, f_1^k \rangle$ for Coverer and f_2 for Disruptor. The outcome of π is the tuple of the k plays generated when Coverer's agents and Disruptor follow their strategies. Formally, $\mathsf{outcome}(\pi) = \langle \rho_1, \ldots, \rho_k \rangle$, where $\rho_i = \mathsf{outcome}(\langle f_1^i, f_2 \rangle)$, for every $i \in [k]$.

We say that a profile π *covers* β iff every objective in β is satisfied in at least one play in $\mathsf{outcome}(\pi)$. That is, if $\cup\{\mathsf{sat}(\rho, \beta) : \rho \in \mathsf{outcome}(\pi)\} = \beta$. Equivalently, for every $j \in [m]$, there exists $i \in [k]$ such that α_j is satisfied in $\mathsf{outcome}(\langle f_1^i, f_2 \rangle)$. A strategy F_1 for Coverer is a *covering* strategy in $\mathcal{G}$ iff for every strategy f_2 for Disruptor, the profile $\langle F_1, f_2 \rangle$ covers β. Then, we say that Coverer wins $\mathcal{G}$ iff she has a covering strategy in $\mathcal{G}$. A strategy f_2 for Disruptor is a *disrupting* strategy in $\mathcal{G}$ iff for every strategy F_1 for Coverer, the profile $\langle F_1, f_2 \rangle$ does not cover β. Then, we say that Disruptor wins $\mathcal{G}$ iff she has a disrupting strategy in $\mathcal{G}$. Finally, we say that a γ-CG $\mathcal{G}$ is *determined* if Coverer has a covering strategy in $\mathcal{G}$ or Disruptor has a disrupting strategy in $\mathcal{G}$.

In the *coverage problem for CGs*, we have to decide whether Coverer has a covering strategy in a given CG. In the *disruption problem for CGs*, we have to decide whether Disruptor has a disrupting strategy in a given CG.

▶ **Example 1.** Consider the Büchi CG $\mathcal{G} = \langle G, 2, \beta \rangle$, with $\beta = \{\alpha_1, \alpha_2, \alpha_3\}$, for the game graph G appearing in Figure 1, and $\alpha_1 = \{u_1, u_2, u_3\}$ (red), $\alpha_2 = \{m_1, u_2, d_3\}$ (green), and $\alpha_3 = \{d_1, d_2, d_3\}$ (yellow). Thus, Coverer should direct two agents in a way that ensures that the three colors are visited infinitely often.

Coverer has a covering strategy in $\mathcal{G}$: In v_0, the tokens of both agents move together, visiting the three colors in a round-robin fashion. If the play stays forever in the left sub-game, β is covered. If Disruptor decides to leave the left sub-game and directs the tokens to v_1 or v_2, the agents split β between them. One agent covers two colors (green and red in u_2, in case the tokens are in v_1, or green and yellow in d_3, in case the tokens are in v_2), and the second agent covers one color (yellow or red, respectively). Thus, β is covered in this case too. ◀

[1] All-Büchi objectives are traditionally referred to as *generalized Büchi*. Then, their dual Exists co-Büchi objectives are referred to as *generalized co-Büchi* [18]. We opt to follow a terminology with All and Exists, making the quantification within the satisfaction requirement clearer.

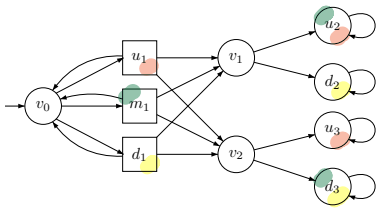

Figure 1 The Büchi CG $\mathcal{G}$.

3 Special Cases of Coverage Games

As a starter, in this section we examine two special cases of CGs: game graphs in which there is only one player, and games in which the number of agents enables an easy reduction to traditional two-player games.

3.1 One-player CGs

We start with one-player CGs, thus when one of the players owns all the vertices. Formally, $\mathcal{G} = \langle G, k, \beta \rangle$, with $G = \langle V_1, V_2, v_0, E \rangle$ is such that $V_2 = \emptyset$ (equivalently, $V = V_1$) or $V_1 = \emptyset$ (equivalently, $V = V_2$). Note that one-player CGs are determined, in the sense that whenever Coverer does not have a covering strategy, Disruptor has an (empty) disrupting strategy, and vice versa.

Consider first the case Coverer has full control. Thus, $G = \langle V, \emptyset, v_0, E \rangle$. It is easy to see that there, covering β amounts to finding k paths in G such that each objective $\alpha_i \in \beta$ is satisfied in at least one path. A path in G is *lasso-shaped* if it is of the form $p \cdot q^\omega$, with $p \in V^*$ and $q \in V^+$. The length of $p \cdot q^\omega$ is defined as $|p| + |q|$.

▶ **Theorem 2.** *The coverage problem for Büchi or co-Büchi CGs with $V = V_1$ is NP-complete.*

Proof. We start with the upper bounds. Consider a CG $\mathcal{G} = \langle G, k, \beta \rangle$ with $V = V_1$. An NP algorithm guesses k lasso-shaped paths in G of length of at most $|V| \cdot |\beta|$, and checks that every objective $\alpha_i \in \beta$ is satisfied in at least one of them. Clearly, if such k paths exist, then Coverer has a covering strategy. Also, checking lasso-shaped paths of length $|V| \cdot |\beta|$ is sufficient. Indeed, for both $\gamma \in \{\text{B,C}\}$, if there exists a path in G that satisfies a subset $\beta' \subseteq \beta$ of γ objectives, then there also exists a cycle in G of length of at most $|V| \cdot |\beta'|$ that visits or avoids all the sets in β'. Since $|\beta'| \leq |\beta|$, the bound follows.

For the lower bound, we describe (easy) reductions from the vertex-cover problem. Consider an undirected graph $G = \langle V, E \rangle$, and $k \leq |V|$. Recall that a vertex cover for G is a set $U \subseteq V$ such that for all edges $\{u, v\} \in E$, we have that $\{u, v\} \cap U \neq \emptyset$. For both $\gamma \in \{\text{B,C}\}$, we construct a γ-CG $\mathcal{G} = \langle G', k, \beta \rangle$ with $V = V_1$ such that Coverer has a covering strategy in $\mathcal{G}$ iff there exists a vertex cover of size at most k in G.

The game graph G' is independent of E and consists of the vertices in V and a new initial vertex v_0. The only edges are self-loops in the vertices in V, and edges from v_0 to all vertices in V. Accordingly, the k agents of Coverer essentially choose k vertices in V. The objectives in β then require these k vertices to be a vertex cover. For $\gamma = \text{B}$, we define $\beta = E$. That is, for every edge $e = \{u, v\} \in E$, the objective e is to visit one of the vertices that correspond to u or v infinitely often. Accordingly, a covering strategy for Coverer induces a vertex cover, and vice versa. For $\gamma = \text{C}$, we define $\beta = \{V \setminus e : e \in E\}$. Since each play ρ is eventually trapped in a self-loop in a vertex in V, the obtained game coincides with the one for $\gamma = \text{B}$.

A detailed description of the reduction and its proof can be found in Appendix A.1. ◀

We continue to the case all the vertices in the game are owned by Disruptor. It is easy to see that there, the fact Coverer has several agents plays no role, as all tokens are going to traverse the same play, chosen by Disruptor. Consequently, searching for a disrupting strategy for Disruptor is easy.

▶ **Theorem 3.** *The disruption problem for Büchi or co-Büchi CGs with $V = V_2$ is NLOGSPACE-complete.*

Proof. Consider a CG $\mathcal{G} = \langle G, k, \beta \rangle$ with $G = \langle \emptyset, V, v_0, E \rangle$. Since Disruptor directs all the tokens together, she has a disrupting strategy iff there is a path in G that satisfies the Exists-$\tilde{\gamma}$ objective dual to β, namely a path that does not satisfy one of the objectives in β. We argue that the latter can be done in NLOGSPACE, for all $\gamma \in \{\text{B,C}\}$. Indeed, an NLOGSPACE algorithm can guess a set $\alpha \in \beta$ and then search for a lasso-shaped path that does not satisfy α. In case $\gamma = \text{B}$, the vertices of the lasso loop should be disjoint from α. In case $\gamma = \text{C}$, the algorithm guesses a vertex $v \in \alpha$ and checks that v appears in the lasso loop. In both cases, it is possible to search for paths of length at most $|V|$. This can be done in NLOGSPACE by guessing the vertices along the path one by one.

Hardness in NLOGSPACE follows from an easy reduction from the non-emptiness problem for nondeterministic ExistsC and ExistsB word automata [28]. ◀

3.2 CGs with one or many agents

Consider a CG $\mathcal{G} = \langle G, k, \beta \rangle$. When $k = 1$, the CG $\mathcal{G}$ is equivalent to the two-player game with the All objective β. Indeed, the single play induced by the strategy of the single agent has to cover all the objectives in β. Thus, the case $k = 1$ can be solved in PTIME, serving as a lower bound for the general case, or more precisely, as a reference point to the study of the complexity for the general case.

When $k \geq |\beta|$, there is no need to assign more than one objective in β to each of the agents. Formally, we have the following.

▶ **Lemma 4.** *Consider a γ-CG $\mathcal{G} = \langle G, k, \beta \rangle$ with $\gamma \in \{B, C\}$ and $k \geq |\beta|$, and the following statements.*

(C1) *Coverer has a covering strategy in $\mathcal{G}$.*
(C2) *For every $\alpha_i \in \beta$, Player 1 wins the two-player γ-game $\mathcal{G}_i = \langle G, \alpha_i \rangle$.*
(C3) *There is $\alpha_i \in \beta$ such that Player 2 wins the two-player γ-game $\mathcal{G}_i = \langle G, \alpha_i \rangle$.*
(C4) *Disruptor has a disrupting strategy in $\mathcal{G}$.*

Then, $(C1)$ iff $(C2)$, $(C3)$ iff $(C4)$, and $(C1)$ and $(C4)$ complement each other.

Proof. First note that since two-player γ-games are determined, then $(C2)$ and $(C3)$ complement each other. Also, as it is impossible for both Coverer to have covering strategy and Disruptor to have a disrupting strategy, $(C1)$ and $(C4)$ contradicts each other. Hence, it is sufficient to prove that $(C2)$ implies $(C1)$ and that $(C3)$ implies $(C4)$.

Assume that $(C2)$ holds. For every $\alpha_i \in \beta$, let f_1^i be a winning strategy for Coverer in $\mathcal{G}_i = \langle G, \alpha_i \rangle$. It is easy to see that $F_1 = \langle f_1^1, \ldots, f_1^{|\beta|}, f_1^{|\beta|+1}, \ldots, f_1^k \rangle$ is a covering strategy for Coverer, for arbitrary strategies $f_1^{|\beta|+1}, \ldots, f_1^k$ for agents $|\beta| + 1, \ldots, k$. Thus, Coverer has a covering strategy, and $(C1)$ holds.

Assume now that $(C3)$ holds. Thus, there is $\alpha_i \in \beta$ such that Disruptor wins the two-player γ-game $\mathcal{G}_i = \langle G, \alpha_i \rangle$. Let f_2 be the winning strategy for Disruptor in $\mathcal{G}_i$. Then, it must be that $\alpha_i \notin \mathsf{sat}(\rho, \beta)$ for every strategy F_1 for Coverer and every play $\rho \in \mathsf{outcome}(\langle F_1, f_2 \rangle)$. Thus, Disruptor has a disrupting strategy and $(C4)$ holds. ◀

By Lemma 4, deciding CGs with $k \geq |\beta|$ agents can be reduced to deciding $|\beta|$ two-player games, thus such CGs are determined and can be decided in PTIME [21, 27, 30].

When $1 < k < |\beta|$, Coverer has to partition the objectives in β among the different agents. As we study in Section 4, this makes the setting much more challenging and interesting.

4 Properties of Coverage Games

In this section we study theoretical properties of CGs. We start with determinacy and show that CGs need not be determined. We then define and study the *decomposability* of objectives in CGs, namely the ability to a-priori partition the objectives among the agents. We argue that decomposability plays a key role in reasoning about CGs.

4.1 CGs need not be determined

Recall that a CG is determined if Coverer has a covering strategy or Disruptor has a disrupting strategy. We show that unlike two-player games, CGs need not be determined. Moreover, undeterminacy holds already for a CG with only two agents and three objectives. Note that this is tight, as CGs with a single agent or only two objectives belong to the $k = 1$ or $k \geq |\beta|$ cases studied in Section 3.2, and are thus determined. Intuitively, the undeterminacy of CGs follows from the incomplete information of Coverer's agents about all the plays in the outcome, as well as the inability of Disruptor to make use of this incomplete information. Formally, we have the following.

▶ **Theorem 5.** *For all $\gamma \in \{B,C\}$, we have that γ-CGs need not be determined. Moreover, undeterminacy holds already for a γ-CG with only two agents and three objectives.*

Proof. For all $\gamma \in \{\text{B,C}\}$, we describe a γ-CG $\mathcal{G} = \langle G, 2, \{\alpha_1, \alpha_2, \alpha_3\}\rangle$ such that neither Coverer has a covering strategy, nor Disruptor has a disrupting strategy. The game graph G appears in Figure 2.

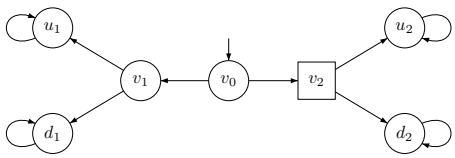

Figure 2 An undetermined coverage game.

When $\gamma = B$, we define $\alpha_1 = \{u_1, u_2\}$, $\alpha_2 = \{d_1, d_2\}$, and $\alpha_3 = \{u_2, d_2\}$. Intuitively (see detailed proof in Appendix A.2), Coverer has no covering strategy as α_3 requires one agent to move to v_2, and no matter how Coverer directs the other agent, Disruptor can move the tokens that reach v_2 so that only one of α_1 and α_2 is satisfied. Also, Disruptor does not have a disrupting strategy, as no matter how the strategy moves token that reach v_2, Coverer can direct the agents so that all objectives are covered.

When $\gamma = C$, we define $\alpha_1 = \{u_1, u_2\}$, $\alpha_2 = \{d_1, d_2\}$, and $\alpha_3 = \{u_1, d_1\}$. Since the only edges from the vertices u_1, d_1, u_2, and d_2 are self-loops, the obtained co-Büchi game coincides with the Büchi game analyzed above, and we are done. ◀

4.2 Decomposability of objectives

A key challenge for Coverer in winning a CG $\mathcal{G} = \langle G, k, \beta\rangle$ with $2 \leq k < |\beta|$ is the need to partition the objectives in β among her k agents. In this section we show that in general, Coverer cannot *a-priori* partition β among the k agents. Moreover, decompositions of β are related to decompositions of G, which are the key to our algorithms for reasoning about CGs.

Consider a game graph $G = \langle V_1, V_2, v_0, E\rangle$. For a vertex $v \in V_1$, let $G^v = \langle V_1, V_2, v, E\rangle$ be G with initial vertex v. Consider a CG $\mathcal{G} = \langle G, k, \beta\rangle$, with $\beta = \{\alpha_1, \ldots, \alpha_m\}$. Let $A = [k]$. For $1 \leq l \leq k$ and a vertex $v \in V_1$, we say that β is (k,l)-*decomposable* in v if Coverer can decompose the task of covering β in G^v by the agents in A to l sub-tasks, each assigned to a different subset of A. Formally, β is (k,l)-decomposable in v if there exists a partition $\beta_1, \ldots, \beta_l$ of β and a partition $A_1, \ldots, A_l$ of A to nonempty sets, such that for every $i \in [l]$, Coverer wins the CG $\mathcal{G}_i^v = \langle G^v, |A_i|, \beta_i\rangle$, namely the game that starts in v and in which the agents in A_i have to cover the objectives in β_i. When $v = v_0$, we say that β is (k,l)-*decomposable* in G.

We start with some easy observations. First, note that, by definition, β is $(k,1)$-*decomposable* in G iff Coverer wins $\mathcal{G}$. Indeed, when $l = 1$, we have that $A_1 = A$ and $\beta_1 = \beta$, thus we do not commit on a decomposition of β, and the definitions coincide. Then, β is (k,k)-*decomposable* in G iff there exists a partition of β to k sets $\beta_1, \ldots, \beta_k$ such that for every $1 \leq i \leq k$, Coverer has a strategy that satisfies the All objective β_i. Indeed, when $l = k$, each of the sets A_i is a singleton, thus the single agent in A_i has to satisfy all the objectives in β_i. Finally, the bigger l is, the more refined is the partition of β to which Coverer commits, making her task harder. Formally, we have the following (see Appendix A.3 for the proof and Example 1 for a CG with no a-priori decomposition):

▶ **Theorem 6.** *For all $\gamma \in \{B, C\}$, we have the following.*

- *Consider a γ-CG $\mathcal{G} = \langle G, k, \beta\rangle$. For every $l \in [k]$ and vertex v of G, if β is (k,l)-decomposable in v, then β is (k,l')-decomposable in v, for all $l' \leq l$.*
- *For every $k \geq 2$, there exists a γ-CG $\mathcal{G} = \langle G, k, \beta\rangle$ with $|\beta| = k+1$ such that Coverer wins $\mathcal{G}$ but β is not (k,l)-decomposable in G, for all $l > 1$.*
- *For every $k \geq 2$ and $1 \leq l < k$, there exists a γ-CG $\mathcal{G} = \langle G, k, \beta\rangle$ such that β is (k,l)-decomposable in G but is not $(k,l+1)$-decomposable in G.*

Consider a vertex $v \in V_1$. For $l \in [k]$, we say that v is a (k,l)-*fork* for β if Coverer has a covering strategy $F_1 = \langle f_1^1, \ldots, f_1^k\rangle$ in G^v such that different agents are sent to l different successors of v. Thus, there exist l successors $v_1, \ldots, v_l$ of v, and for all $i \in [l]$, there is at least one agent $j \in [k]$ such that $f_1^j(v) = v_i$. Note that we do not require f_1^i to be memoryless and refer here to the strategy in the first round in G^v. The following lemma then follows from the definitions.

▶ **Lemma 7.** *Consider a CG $\mathcal{G} = \langle G, k, \beta\rangle$. For every vertex $v \in V_1$ of G and $l \in [k]$, if v is a (k,l)-fork for β, then β is (k,l)-decomposable in v.*

Proof. Consider a vertex $v \in V_1$ and $1 \leq l \leq k$ such that v is a (k,l)-fork for β. Let $F_1 = \langle f_1^1, \ldots, f_1^k\rangle$ be a winning strategy for Coverer that sends the agents to l different successors $v_1, \ldots, v_l$ of v. For every $1 \leq i \leq l$, let A_i be the set of agents $j \in [k]$ such that $f_1^j(v) = v_i$, and let $F_1^i = \{f_1^j\}_{j \in A_i}$ be the set of strategies of the agents sent to v_i. Let β_i be the set objectives of β that F_1^i covers. That is, $\alpha_j \in \beta_i$ iff for all strategies f_2 of Disruptor, there is a play in $\mathsf{outcome}(F_1^i, f_2)$ that satisfies α_j. Equivalently, $\beta_i \subseteq \beta$ is the maximal subset of β such that F_1^i is a winning strategy for Coverer in $\langle G_{v_i}, |A_i|, \beta_i\rangle$. It is then sufficient to show that $\bigcup_{i \in [l]} \beta_i = \beta$. Assume by contradiction that there exists an objective $\alpha_j \in \beta$ such that $\alpha_j \notin \beta_i$, for every $i \in [l]$, which implies there exists a strategy f_2^i for Disruptor in G_{v_i} such that α_j is not covered in $\mathsf{outcome}(\langle F_1^i, f_2^i\rangle)$. Let f_2 be the strategy for Disruptor from v that follows f_2^i from v_i, for every $i \in [l]$. It is easy to see that α_j is not covered in $\mathsf{outcome}(\langle F_1, f_2\rangle)$, contradicting the fact F_1 is a winning strategy. ◀

We say that a vertex $v \in V_1$ is a *fork* if it is a (k, l)-fork for β, for some $2 \leq l \leq k$. We denote by $F \subseteq V_1$ the set of vertices that are forks. Let $V_{avoid} \subseteq V$ be the set of vertices from which Disruptor can avoid reaching forks. That is, $v \in V_{avoid}$ if there exists a strategy f_2 for Player 2 from v such that for every strategy f_1 for Player 1, the play $\mathsf{outcome}(\langle f_1, f_2 \rangle)$ does not reach F. Finally, let $G_{avoid} = \langle V_1', V_2', v_0, E' \rangle$ be the sub-graph of G with vertices in V_{avoid}. That is, $V_1' = V_1 \cap V_{avoid}$, $V_2' = V_2 \cap V_{avoid}$, and $E' = E \cap (V_{avoid} \times V_{avoid})$.

▶ **Example 8.** Consider the Büchi CG $\mathcal{G}$ from Example 1. Recall that Coverer has a covering strategy in $\mathcal{G}$. We claim that the only forks for β in G are v_1 and v_2. Thus, a winning strategy for Coverer cannot a-priori decompose the objectives in β. Note that this is the case also for the covering strategy described in Example 1.

Indeed, in v_1, Coverer can assign both α_1 and α_2 to an agent that proceeds to u_2 and assign α_3 to an agent that proceeds to d_2. Similarly, in v_2, Coverer can assign α_1 to an agent that proceeds to u_3 and assign both α_2 and α_3 to an agent that proceeds to d_3. Also, as Disruptor may direct tokens to either v_1 or v_2, and the above decompositions are different and are the only possible decompositions in v_1 and v_2, Coverer cannot a-priori decompose β. Hence, v_0 is not a fork. Finally, as β is not covered in each of u_2, d_2, u_3, and d_3, they are not forks either.

Note also that since Disruptor can direct tokens from u_1, m_1, and d_1 back to v_0, the graph G_{avoid} is the sub-graph of G whose vertices are v_0, u_1, m_1 and d_1.

▶ **Theorem 9.** *Consider a CG $\mathcal{G} = \langle G, k, \beta \rangle$, and a vertex $v \in V$. Then, Coverer has a covering strategy in $\langle G^v, k, \beta \rangle$ iff $v \notin V_{avoid}$, or $v \in V_{avoid}$ and Coverer wins $\langle G^v_{avoid}, 1, \beta \rangle$.*

Proof. We first prove that for all vertices $v \notin V_{avoid}$, Coverer has a strategy to cover β from v. Consider a vertex $v \notin V_{avoid}$. By the definition of V_{avoid}, Coverer has a strategy to reach a fork from v. Hence, Coverer can cover β from v by letting all agents follow the same strategy until some (k, l)-fork u for β is reached. By Lemma 7, β is (k, l)-decomposable in u. Hence, once the k tokens of the agents reach u, Coverer can cover β by decomposing it. Let F_1^v denote a covering strategy for Coverer from a vertex $v \notin V_{avoid}$.

We argue that for every vertex $v \in V_{avoid}$, Coverer has a strategy to cover β from v iff Coverer wins $\langle G^v_{avoid}, 1, \beta \rangle$. Consider a vertex $v \in V_{avoid}$, and assume first that Coverer has a covering strategy from v. Since Disruptor can force the play to stay in G_{avoid}, every covering strategy for Coverer from v in G is also a covering strategy from v in G_{avoid}. Also, for every covering strategy $F_1 = \langle f_1^1, \ldots, f_1^k \rangle$ from v, the strategies for the different agents agree with each other as long as the play stays in G_{avoid}. That is, $f_1^1(h) = \cdots = f_1^k(h)$, for every $h \in V^*_{avoid} \cdot V_1'$. Indeed, vertices from which there exists a covering strategy for Coverer that sends different agents to different successors are forks, and by the definition of V_{avoid}, the sub-graph G_{avoid} does not contain forks. Hence, the strategy $f_1 : V^*_{avoid} \cdot V_1' \to V_{avoid}$ with $f_1(h) = f_1^1(h)$ for every $h \in V^*_{avoid} \cdot V_1'$ is a covering strategy for Coverer $\langle G^v_{avoid}, 1, \beta \rangle$.

For the second direction, assume that Coverer wins $\langle G^v_{avoid}, 1, \beta \rangle$. We show that Coverer has a covering strategy from v. Let f_1 be a covering strategy for Coverer in $\langle G^v_{avoid}, 1, \beta \rangle$. Consider the strategy $F_1 = \langle f_1^1, \ldots, f_1^k \rangle$ in which, for all $1 \leq i \leq k$, the strategy f_1^i agrees with f_1 as long as the play stays in G_{avoid}, and proceeds with F_1^u once the play (that is, all tokens together, as this may happen only in a transition taken by Disruptor), leaves G_{avoid} and reaches a vertex $u \notin V_{avoid}$. For every strategy f_2 for Disruptor, either all the outcomes of F_1 and f_2 stay in G_{avoid}, in which case β is covered by each of the agents, or all the outcomes leave G_{avoid}, in which case they follow strategies that cover β by decomposing it. ◀

5 The Complexity of the Coverage Problem

In this section, we study the complexity of deciding the existence of covering strategies in Büchi and co-Büchi CGs, and show that the problem is PSPACE-complete.

We start with upper bounds. Consider a CG $\mathcal{G} = \langle G, k, \beta \rangle$. Recall the set F of forks for β, and the game graph G_{avoid}, in which Disruptor can avoid reaching forks. By Theorem 9, Coverer has a covering strategy in G iff Player 1 has a winning strategy in G_{avoid} for the All objective β, and when the play leaves G_{avoid} (or if the initial vertex is not in G_{avoid}), Coverer can cover β by reaching a fork, from which β can be decomposed.

The above characterization is the key to our algorithm for deciding whether Coverer has a covering strategy. Essentially, the algorithm guesses the set U of forks, checks that Player 1 wins the game with the All objective β in the sub-graph induced by U, and checks (recursively) that the vertices in U are indeed forks. The checks and the maintenance of the recursion require polynomial space. Since NPSPACE=PSPACE, we have the following. [2]

▶ **Theorem 10.** *The coverage problem for Büchi or co-Büchi CGs can be solved in PSPACE.*

Proof. We describe a non-deterministic Turing machine (NTM) T that runs in polynomial space, such that T accepts a CG $\mathcal{G}$ iff Coverer has a covering strategy in $\mathcal{G}$.

Given a CG $\mathcal{G} = \langle G, k, \beta \rangle$, the NTM T guesses a set of vertices $U \subseteq V$, and checks that they are forks. To check that a vertex $u \in U$ is a fork, the NTM guesses a partition $\beta_1, \ldots, \beta_l$ of β and a partition $A_1, \ldots, A_l$ of $[k]$ to nonempty sets, for some $2 \leq l \leq k$, and checks recursively whether Coverer has a covering strategy in $\langle G^u, |A_i|, \beta_i \rangle$, for every $i \in [l]$. If one of the recursive checks fails, the NTM rejects. Otherwise, T calculates the restriction G' of G to vertices from which Disruptor can force the play to avoid U. Then, T accepts if the initial vertex v of G is not in G' or Player 1 wins from v in the All-γ game $\langle G', \beta \rangle$. Otherwise, T rejects.

We prove the correctness of the construction. That is, we prove that T accepts a game $\mathcal{G}$ iff Coverer has a covering strategy in $\mathcal{G}$. By Theorem 9, Coverer has a covering strategy from a vertex v iff $v \notin V_{avoid}$, or $v \in V_{avoid}$ and Coverer wins $\langle G^v_{avoid}, 1, \beta \rangle$. So, if Coverer has a covering strategy, the NTM can guess the set F of forks for the current set β of objectives and number k of agents. Then, G' coincides with G_{avoid}, and by Theorem 9, the NTM accepts. For the second direction, assume the NTM accepts. Then, the vertices in U are forks by definition, and Coverer wins $\langle G', 1, \beta \rangle$. By Theorem 9, it is enough to show that if $v \in G_{avoid}$, then Coverer wins $\langle G^v_{avoid}, 1, \beta \rangle$. Since the vertices in U are forks, G_{avoid} is a sub-graph of G'. So, since Disruptor can force the play to stay in G_{avoid} if v is in it, we also have that Coverer wins $\langle G^v_{avoid}, 1, \beta \rangle$.

To complete the proof, we show that the NTM runs in polynomial space. Note that only a polynomial number of recursive checks can be made before reaching an empty set of objectives, and each check requires a polynomial space for guessing U and the appropriate partitions. Also, calculating G' and checking whether Player 1 wins an All-γ game can be done in polynomial time, for both $\gamma \in \{\text{B,C}\}$ [27]. ◀

Note that the algorithm can return a covering strategy (if one exists). Such a strategy can be described by the set F of forks, the sub-game G_{avoid}, a covering strategy in $\langle G_{avoid}, 1, \beta \rangle$, and the strategies for reaching forks. Then, for each fork, the strategy describes the partitions

[2] It is not hard to see that the proof of Theorem 10 applies to all prefix-independent objective types γ such that All-γ games can be decided in PSPACE.

of β and $[k]$, and the (recursive) description of the strategies in the decomposed game. Note that the description may require more than polynomial space.

We continue to the lower bounds.

▶ **Theorem 11.** *The coverage problem for Büchi or co-Büchi CGs is PSPACE-hard.*

Proof. We describe reductions from QBF. We start with Büchi CGs. Consider a set of variables $X = \{x_1, \ldots, x_n\}$ and a QBF formula $\Phi = Q_1 x_1, \ldots, Q_n x_n \varphi$ where $\varphi = C_1 \wedge \cdots \wedge C_m$, with $C_i = (l_i^1 \vee l_i^2 \vee l_i^3)$, for every $i \in [m]$. We construct a Büchi CG $\mathcal{G} = \langle G, |X|, \beta \rangle$ such that $\Phi = \textbf{true}$ iff Coverer has a covering strategy in $\mathcal{G}$.

The game graph G (see example in Figure 3) lets the players choose an assignment to the variables in X, following the quantification order of Φ and starting from the outermost variable. Starting from the initial vertex, when the play reaches a vertex that corresponds to an existential variable x_i, Coverer chooses between choosing an assignment to x_i, which involves proceeding to a self-looped sink that corresponds to the literal x_i, or proceeding to a self-looped sink that corresponds to the literal $\overline{x_i}$, and proceeding to the next variables. When the play reaches a vertex that corresponds to a universal variable x_i, Disruptor chooses an assignment to x_i, by proceeding to a vertex that corresponds to the literal x_i, and proceeding to a vertex that corresponds to the literal $\overline{x_i}$. In those vertices, Coverer chooses between staying in the current vertex, and proceeding to the next variables.

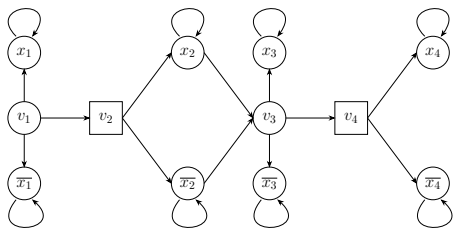

Figure 3 The game graph G for $\Phi = \exists x_1 \forall x_2 \exists x_3 \forall x_4 \varphi$.

Since the vertices that correspond to existential variables are self-looped sinks, the choices of Coverer are not reflected in the history of plays that reach the subsequent variables. Hence, when Disruptor chooses an assignment to a universal variable, she is unaware of the assignments Coverer chose for preceding existential variables. The vertices that correspond to universal variables are not sinks, thus when choosing assignments to existential variables, Coverer is aware of the assignments Disruptor chose for preceding universal variables.

Then, for every variable $x \in X$, we define an objective α_x that forces Coverer to allocate an agent to stay in the vertices that correspond to the assignment chosen for x. Also, for every clause C_i, we define the objective α_{C_i} that consists of vertices that correspond to C_i's literals. Thus, the joint assignment satisfies φ iff Coverer can cover all the objectives. See Appendix A.4 for a formal description of $\mathcal{G}$ and the correctness proof.

For co-Büchi CGs, note that, as in previous reductions, a Büchi objective $\alpha \subseteq (X \cup \overline{X})$ is satisfied in a play ρ iff the co-Büchi objective $(X \cup \overline{X}) \setminus \alpha$ is satisfied in ρ, and thus the same reduction with dual objectives applies. ◀

6 The Complexity of the Disruption Problem

In this section, we study the complexity of deciding the existence of disrupting strategies in Büchi and co-Büchi CGs. Note that since CGs are undetermined, the coverage and disruption problems are not dual, and so the results in Section 5 do not imply that the

disruption problem is PSPACE-complete. In fact, we show here that the disruption problem is Σ_2^{P}-complete. Thus, at least in terms of its theoretical complexity class, it is easier than the coverage problem.

We start with upper bounds. Consider a CG $\mathcal{G} = \langle G, k, \beta \rangle$, and consider a two-player game played on G. For every strategy f_2 for Player 2, let $\Delta_{f_2} \subseteq 2^{\beta}$ be the set of maximal subsets $\delta \subseteq \beta$ such that Player 1 has a strategy that ensures the satisfaction of the All objective δ when Player 2 uses f_2. That is, Δ_{f_2} is the set of maximal sets $\delta \subseteq \beta$ for which there exists a strategy f_1 for Player 1 such that $\mathsf{sat}(\mathsf{outcome}(\langle f_1, f_2 \rangle), \beta) = \delta$.

▶ **Lemma 12.** *Consider a strategy f_2 for Player 2. For every strategy f_1 for Player 1, there exists $\delta \in \Delta_{f_2}$ such that $\mathsf{outcome}(\langle f_1, f_2 \rangle)$ satisfies the All-$\tilde{\gamma}$ objective $(\beta \setminus \delta)$.*

Proof. Consider a strategy f_2 for Disruptor, and a strategy f_1 for Coverer, and let $\rho = \mathsf{outcome}(\langle f_1, f_2 \rangle)$. By the definition of Δ_{f_2}, there exists $\delta \in \Delta_{f_2}$ such that $\mathsf{sat}(\rho, \beta) \subseteq \delta$. Accordingly, $(\beta \setminus \delta) \cap \mathsf{sat}(\rho, \beta) = \emptyset$, and so ρ satisfies all the $\tilde{\gamma}$ objectives in $\beta \setminus \delta$. ◀

Back to CGs, the intuition behind the definition of Δ_{f_2} is that $\delta \in \Delta_{f_2}$ iff whenever Disruptor follows f_2, Coverer can allocate one agent that covers exactly all the objectives in δ. Formally, we have the following, which follows directly from the definitions.

▶ **Lemma 13.** *Consider a CG $\mathcal{G} = \langle G, k, \beta \rangle$. A strategy f_2 for Disruptor is a disrupting strategy in $\mathcal{G}$ iff for every k sets $\delta_1, \ldots, \delta_k \in \Delta_{f_2}$, we have that $\bigcup_{i \in [k]} \delta_i \neq \beta$.*

We use the observations about Δ_{f_2} in order to restrict the search for disrupting strategies. We start with Büchi CGs.

▶ **Theorem 14.** *Disruptor has a disrupting strategy in a Büchi CG $\mathcal{G}$ iff she has a memoryless disrupting strategy in $\mathcal{G}$.*

Proof. Consider a Büchi CG $\mathcal{G} = \langle G, k, \beta \rangle$. Assume Disruptor has a disrupting strategy f_2 in $\mathcal{G}$. By Lemma 12, for every strategy f_1 for an agent of Coverer, there exists $\delta \in \Delta_{f_2}$ such that $\mathsf{outcome}(\langle f_1, f_2 \rangle)$ satisfies the AllC objective $\beta \setminus \delta$, which is equivalent to the co-Büchi objective $\cup(\beta \setminus \delta)$. Accordingly, f_2 is a winning strategy for Player 2 in a game played on G in which her objective is the ExistsC objective $\{\cup(\beta \setminus \delta)\}_{\delta \in \Delta_{f_2}}$. Let f_2' be a memoryless winning strategy for Player 2 in this game. Since ExistsC objectives require memoryless strategies, such a strategy exists. In Appendix A.5, we prove that f_2' is a disrupting strategy in $\mathcal{G}$. ◀

▶ **Theorem 15.** *The disruption problem for Büchi CGs is in Σ_2^{P}.*

Proof. Consider a Büchi CG $\mathcal{G} = \langle G, k, \beta \rangle$. An NP algorithm that uses a co-NP oracle guesses a memoryless strategy f_2 for Disruptor, and then checks that Coverer does not have a covering strategy in the one-player game defined over the sub-graph G_{f_2} induced by f_2. That is, the sub-graph of G in which the edges from vertices in V_2 agree with f_2. Since Coverer essentially owns all the vertices in G_{f_2}, then by Theorem 2, deciding the existence of covering strategies in G_{f_2} can be done in NP, implying the Σ_2^{P}-upper bound. ◀

We continue to co-Büchi CGs. Here, disrupting strategies need not be memoryless, and in fact need not be polynomial. Accordingly, the proof is more complicated and is based on a symbolic description of disrupting strategies.

A *superset objective* is a set $\mathcal{F} \subseteq 2^{2^V}$ of AllB objectives, and it is satisfied in a play ρ iff at least one AllB objective in $\mathcal{F}$ is satisfied in ρ [16]. By [16], every superset objective has an equivalent AllB objective. Accordingly, in two-player superset games, Player 2 has

an ExistsC objective, thus Player 2 wins iff she has a memoryless winning strategy. Since a memoryless strategy for Player 2 can be checked in polynomial time and the game is determined, the problem of deciding whether there exists a winning strategy for a superset objective is co-NP-complete [16].

We start with an easy characterization of winning strategies for AllB objectives.

▶ **Lemma 16.** *Consider an AllB game $\mathcal{G} = \langle G, \delta \rangle$, with $\delta = \{\alpha_1, \ldots, \alpha_m\}$. Player 1 wins $\mathcal{G}$ iff there exist a set $W \subseteq V$ of vertices and a set $\{g_1^i : i \in [m]\}$ of memoryless strategies for Player 1, such that Player 1 can force the play to reach W, and for every vertex $v \in W$ and $i \in [m]$, the strategy g_1^i forces each play from v to reach $\alpha_i \cap W$, while staying in W.*

We say that W is a *winning cage* for δ, and that $\{g_1^i : i \in [m]\}$ are its *witnesses*.

Lemma 17 below suggests a way to describe winning strategies for AllB objectives by the sets of successors a winning strategy chooses infinitely often from every vertex.

For a set of vertices $U \subseteq V$, a strategy f_1 for Player 1 is a *U-trap* if it forces the play to reach and stay in U. That is, if $\mathit{inf}(\mathsf{outcome}(\langle f_1, f_2 \rangle)) \subseteq U$, for every strategy f_2 for Player 2.

A *fairness requirement* for Player 1 is a function $g : V_1 \to 2^V$ that maps each vertex $v \in V_1$ to a subset of its successors. We say that a strategy f_1 for Player 1 *respects* g if for every vertex $v \in V_1$, the strategy f_1 only chooses successors for v from $g(v)$, and when v is visited infinitely often, then f_1 proceeds to each of the successors of v in $g(v)$ infinitely often. That is, for every vertex $v \in V_1$ and a prefix of a play $h \in V^*$ we have that $f_1(h \cdot v) \in g(v)$, and for every strategy f_2 for Player 2 and a vertex $v \in V_1$, if $v \in \mathit{inf}(\mathsf{outcome}(\langle f_1, f_2 \rangle))$, then $g(v) \subseteq \mathit{inf}(\mathsf{outcome}(\langle f_1, f_2 \rangle))$.

▶ **Lemma 17.** *Consider an AllB game $\mathcal{G} = \langle G, \delta \rangle$, with $\delta = \{\alpha_1, \ldots, \alpha_m\}$. Player 1 wins $\mathcal{G}$ iff there exists a set of vertices $U \subseteq V$ and a fairness requirement g for Player 1 such that every U-trap strategy for Player 1 that respects g is a winning strategy for Player 1 in $\mathcal{G}$, and such a strategy exists.*

Proof. First, if there is a U-trap strategy for Player 1 that respects a fairness requirement g such that every such strategy is a winning strategy in $\mathcal{G}$, then Player 1 has a winning strategy in $\mathcal{G}$.

For the other direction, assume Player 1 wins $\mathcal{G}$. Consider a minimal winning cage $W \subseteq V$ and witnesses $\{g_1^i : i \in [m]\}$ for W. Note that by definition, Player 1 has a W-trap strategy that respects a fairness requirement g with $g(v) = \{g_1^i(v) : i \in [m]\}$ for every $v \in V_1 \cap W$. We show that every such strategy is a winning strategy for Player 1 in $\mathcal{G}$.

Consider a strategy f_1^* for Player 1 in $\mathcal{G}$ that forces the play to reach W, and, starting with $i = 1$, follows g_1^i until the play reaches α_i, where it switches to follow $g_1^{(i+1) \mod m}$. It is easy to see that f_1^* is a winning strategy for Player 1 in $\mathcal{G}$.

Now, consider a strategy f_1 for Player 1 that is a W-trap, and respects g. Note that such a strategy exists, since W is a winning cage for δ and for every vertex $v \in V_1 \cap W$, we have that $g(v) \subseteq W$. We show that f_1 is a winning strategy for Player 1 in $\mathcal{G}$.

Assume by contradiction otherwise. Thus, there exists a strategy f_2 for Player 2 such that $U = \mathit{inf}(\mathsf{outcome}(\langle f_1, f_2 \rangle))$ does not intersect with all the sets in δ. That is, there exists $i \in [m]$ such that $U \cap \alpha_i = \emptyset$. We show that there exists a strategy f_2' for Player 2 with $U' = \mathit{inf}(\mathsf{outcome}(\langle f_1^*, f_2' \rangle))$ such that $U' \subseteq U$, thus $U' \cap \alpha_i = \emptyset$, which contradicts the fact that f_1^* is a winning strategy for Player 1 in $\mathcal{G}$.

Let f_2' be a strategy for Player 2 with which $\mathsf{outcome}(\langle f_1^*, f_2' \rangle)$ reaches U, and chooses for every vertex $v \in U \cap V_2$ a successor in U. Note that Player 2 can make the play reach U when Player 1 uses f_1^*, as otherwise f_1^* is a strategy that forces the play to avoid U, which

contradicts the fact W is minimal. Note that $U' \subseteq U$, since both strategies choose successors in U for vertices in U. Indeed, by the definition of f_1^*, for every vertex $v \in U' \cap V_1$ the set of successors f_1^* chooses for v is a subset of $g(v) = \{g_1^i(v) : i \in [m]\}$. To conclude, note that by the definition of f_1, we have that $g(v) \subseteq U$. Indeed, every vertex $v \in U \cap V_1$ is visited infinitely often in $\mathsf{outcome}(\langle f_1, f_2\rangle)$, thus all the vertices in $g(v)$ are visited infinitely often. ◀

By Lemma 17, we do not need to guess an explicit strategy for Player 1 in order to determine whether she wins an AllB or a superset game. Instead, we can guess a set of vertices that might appear infinitely often, and the set of successors we choose infinitely often from each of those vertices.

Back to CGs, we show that disrupting strategies can be described symbolically using a fairness requirement.

We say that a set of subsets of objectives $\Delta \subseteq 2^\beta$ is *k-wise intersecting* if the intersection of every k sets in Δ is not empty. That is, $\bigcap_{i\in[k]} \delta_i \neq \emptyset$, for every $\delta_1, \ldots, \delta_k \in \Delta$.

▶ **Lemma 18.** *A strategy f_2 for Disruptor is a disrupting strategy iff the set $\{(\beta \setminus \delta) : \delta \in \Delta_{f_2}\}$ is k-wise intersecting.*

Proof. Assume first that f_2 is a disrupting strategy, and consider k sets $\delta_1, \ldots, \delta_k \in \Delta_{f_2}$. By the definition of Δ_{f_2}, there exist strategies $f_1^1, \ldots, f_1^k$ for the agents of Coverer with $\mathsf{sat}(\mathsf{outcome}(\langle f_1^i, f_2\rangle), \beta) = \delta_i$, for every $i \in [k]$. Since f_2 is a disrupting strategy, then, by Lemma 13, we have that $\bigcup_{i\in[k]} \delta_i \neq \beta$. Therefore, $\bigcap_{i\in[k]}(\beta \setminus \delta_i) \neq \emptyset$, and so $\{(\beta \setminus \delta) : \delta \in \Delta_{f_2}\}$ is k-wise intersecting.

For the second direction, consider a strategy f_2 for Disruptor such that $\{(\beta \setminus \delta) : \delta \in \Delta_{f_2}\}$ is k-wise intersecting. Then, for every strategy $F_1 = \langle f_1^1, \ldots, f_1^k\rangle$ for Coverer and $\delta_1, \ldots, \delta_k \in \Delta_{f_2}$ with $\mathsf{sat}(\mathsf{outcome}(f_1^i, f_2)) \subseteq \delta_i$, for every $i \in [k]$, there exists $\alpha_j \in \bigcap_{i\in[k]}(\beta \setminus \delta_i)$. Thus, α_j is not covered in $\mathsf{outcome}(\langle F_1, f_2\rangle)$. Accordingly, f_2 is a disrupting strategy. ◀

▶ **Lemma 19.** *If Disruptor has a disrupting strategy in a co-Büchi CG, then there exist a set of vertices U and a fairness requirement g such that every U-trap strategy for Disruptor that respects g is a disrupting strategy, and such a strategy exists.*

Proof. Consider a co-Büchi CG $\mathcal{G} = \langle G, k, \beta\rangle$, and assume there exists a disrupting strategy f_2 for Disruptor. Let δ_{f_2} be the AllB objective equivalent to the superset objective $\{(\beta \setminus \delta) : \delta \in \Delta_{f_2}\}$. Since f_2 is a winning strategy for the superset objective $\{(\beta \setminus \delta) : \delta \in \Delta_{f_2}\}$, and thus for its equivalent AllB objective δ_{f_2}, then, by Lemma 17, there exist a set of vertices U and a fairness requirement g such that every U-trap strategy f_2' for Disruptor that respects g is a winning strategy for δ_{f_2}, and hence also a winning strategy for the superset objective $\{(\beta \setminus \delta) : \delta \in \Delta_{f_2}\}$. Also, such a strategy exists. By Lemma 18, in order to show that every such strategy f_2' is a disrupting strategy, it is enough to show that $\{(\beta \setminus \delta) : \delta \in \Delta_{f_2'}\}$ is k-wise intersecting.

Consider a U-trap strategy f_2' that respects g. Note that since f_2' is a winning strategy for the superset objective $\{(\beta \setminus \delta) : \delta \in \Delta_{f_2}\}$, for every set $\delta \in \Delta_{f_2'}$, there exists a set $\delta' \in \Delta_{f_2}$ such that $(\beta \setminus \delta') \subseteq (\beta \setminus \delta)$. Now, consider k sets $\delta_1, \ldots, \delta_k \in \Delta_{f_2'}$ of objectives, and for every $i \in [k]$, let $\delta_i' \in \Delta_{f_2}$ be a set of objectives such that $(\beta \setminus \delta_i') \subseteq (\beta \setminus \delta_i)$. Since $\bigcap_{i\in[k]}(\beta \setminus \delta_i') \subseteq \bigcap_{i\in[k]}(\beta \setminus \delta_i)$ and $\{(\beta \setminus \delta) : \delta \in \Delta_{f_2}\}$ is k-wise intersecting, we have that $\bigcap_{i\in[k]}(\beta \setminus \delta_i) \neq \emptyset$. Accordingly, $\{(\beta \setminus \delta) : \delta \in \Delta_{f_2'}\}$ is k-wise intersecting. ◀

The fact we can restrict attention to disrupting strategies that are given by fairness requirements leads to an upper bound for the disruption problem.

▶ **Theorem 20.** *The disruption problem for co-Büchi CGs is in* Σ_2^{P}.

Proof. An NP algorithm that uses a co-NP oracle guesses a set of vertices $U \subseteq V$ and a fairness requirement $g : (V_2 \cap U) \to (2^U \setminus \{\emptyset\})$, checks that there exists a U-trap strategy for Disruptor that respects g, and then the co-NP oracle checks that every U-trap strategy for Disruptor that respects g is a disrupting strategy. By Lemma 19, the search can be restricted to trap strategies that respect a given fairness requirement. In Appendix A.6 we describe how the above two steps can indeed be done in NP and co-NP. In particular, checking the U-traps that respect g is based on the fact that for every strategy for an agent of Coverer, the set of vertices visited infinitely often in the resulted play are strongly connected subsets of U that contain all the successors of vertices in V_2 according to g, and the rest of the vertices are successors of vertices in V_1. ◀

We continue to lower bounds. Note that the game in the reduction in the proof of Theorem 11 is determined when Φ is of the form $\forall X \exists Y \varphi$. Indeed, in this case the assignment to the universal variables is independent of the assignment to the existential variables, and thus if $\Phi = \textbf{false}$, which holds iff the 2QBF formula $\exists X \forall Y \overline{\varphi}$ is valid, Disruptor has a disrupting strategy. Since 2QBF is Σ_2^{P}-hard, we have the following (see full proof in Appendix A.7).

▶ **Theorem 21.** *The disruption problem for Büchi or co-Büchi CGs is* Σ_2^{P}*-hard.*

7 Discussion

We introduced and studied coverage games, extending multi-agent planning to adversarial factors that go beyond those studied in existing frameworks.

Below we discuss variants of CGs. Clearly, all classical extensions to the underlying two-player game (e.g., concurrency, probability, partial visibility, etc.) and to the objectives (e.g., richer winning conditions, weighted objectives, mixing objectives of different types, etc.), as well as classical extensions from planning (e.g., dynamic objectives, optimality of agents and their resources, etc.) are interesting also in the setting of CGs. For example, handling *reachability* objectives, it is interesting to search for strategies of Coverer that reach the objectives via short paths, and one may optimize the longest or the average path. Although the extension to richer objectives is an obvious direction for future work, we note that the study of Büchi and co-Büchi better highlights the differences between CGs and standard games, as the complexity is not dominated by challenges that have to do with the objective. For example, by Footnote 2, the extension of the general Müller objective to CGs involves no cost, whereas for the Büchi and co-Büchi objectives, we exhibit a variety of complexity classes, obtained for the different variants of CGs. As our results show, the transition from multiple objectives (a single agent) to CGs adds the challenge of decomposing the set β of objectives among the agents. For example, while multiple co-Büchi objectives can be merged, making them easier than multiple Büchi objectives, co-Büchi CGs are not easier than Büchi CGs. It is interesting to study how the need to decompose the objectives affects the problem in richer settings.

More interesting in our context are extensions that have to do with the operation of several agents: communication among the agents and the ability of Disruptor to also use different agents. Recall that in the CGs studied here, the strategy of each agent may depend on the history of her interaction with Disruptor, but is independent of the interaction of the other agents. Settings in which the agents communicate with each other corresponds to CGs in which their strategies depend on the history of all interactions. Clearly, communication

can help Coverer. For example, if two drones are tasked with patrolling an area and Drone 1 is drifting northward at a certain location, knowing this could prompt Drone 2 to avoid that location or to fly southward. Also, beyond full information about the other agents, various applications induce interesting special cases, such as visibility of a subset of the agents, of agents in some radius, of their current location only, and so on.

As for a richer disruption, recall that in the CGs studied here, all the agents of Coverer interact with the same strategy of Disruptor. It is interesting to consider settings in which Disruptor also operates a number l of agents. Then, a strategy for Disruptor is a vector $F_2 = \langle f_2^1, \ldots, f_2^l \rangle$ of strategies, and Coverer should cover all the objectives in β for every F_2 and for every possible pairing of each of her agents with those of Disruptor. This extension corresponds to settings in which different agents may face different responses, even after the exact same interaction. Clearly, this can help Disruptor. For example, if a system aims to ensure that some road remains uncongested, it may limit the number of cars in each zone by directing different cars to different directions. From a technical point of view, the setting involves additional challenges. In particular, decomposition now has an additional parameter. For example, it can be shown that if $l = k$, then Coverer can cover β iff there is an a-priori decomposition of β among the agents.

References

1. L. Alon, N. Agmon, and G.A Kaminka. Taking turns in complete coverage for multiple robots. In *Distributed Autonomous Robotic Systems*, pages 401–412. Springer International Publishing, 2019.
2. N. Bertrand, M. Dewaskar, B. Genest, H. Gimbert, and A.A. Godbole. Controlling a population. *Logical Methods in Computer Science*, 15(3):1–30, 2019. doi:10.23638/LMCS-15(3:6)2019.
3. R. Bloem, K. Chatterjee, and B. Jobstmann. Graph games and reactive synthesis. In *Handbook of Model Checking.*, pages 921–962. Springer, 2018. doi:10.1007/978-3-319-10575-8_27.
4. P. Bouyer, R. Brenguier, N. Markey, and M. Ummels. Concurrent games with ordered objectives. In *Proc. 15th Int. Conf. on Foundations of Software Science and Computation Structures*, volume 7213 of *Lecture Notes in Computer Science*, pages 301–315. Springer, 2012. doi:10.1007/978-3-642-28729-9_20.
5. K. Chatterjee, L. Doyen, T.A. Henzinger, and J-F. Raskin. Generalized mean-payoff and energy games. In *Proc. 30th Conf. on Foundations of Software Technology and Theoretical Computer Science*, volume 8 of *LIPIcs*, pages 505–516. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2010. doi:10.4230/LIPICS.FSTTCS.2010.505.
6. K. Chatterjee, W. Dvorak, M. Henzinger, and V. Loitzenbauer. Conditionally optimal algorithms for generalized büchi games. In *41st Int. Symp. on Mathematical Foundations of Computer Science*, volume 58 of *LIPIcs*, pages 25:1–25:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPICS.MFCS.2016.25.
7. K. Chatterjee, T. A. Henzinger, and N. Piterman. Generalized parity games. In *Proc. 10th Int. Conf. on Foundations of Software Science and Computation Structures*, volume 4423 of *Lecture Notes in Computer Science*, pages 153–167. Springer, 2007. doi:10.1007/978-3-540-71389-0_12.
8. K. Chatterjee, Z. Komárková, and J. Kretínský. Unifying two views on multiple mean-payoff objectives in markov decision processes. In *Proc. 30th ACM/IEEE Symp. on Logic in Computer Science*, pages 244–256, 2015.
9. S.W. Cho, H.J. Park, H. Lee, D.H. Shim, and S.Y. Kim. Coverage path planning for multiple unmanned aerial vehicles in maritime search and rescue operations. *Computers and Industrial Engineering*, 161:107612, 2021. doi:10.1016/j.cie.2021.107612.
10. H. Choset. Coverage for robotics - a survey of recent results. *Annals of Mathematics and Artificial Intelligence*, 31:113–126, 2001. doi:10.1023/A:1016639210559.

11 A.G. Clark, N. Walkinshaw, and R.M. Hierons. Test case generation for agent-based models: A systematic literature review. *Information and Software Technology*, 135:106567, 2021. doi:10.1016/J.INFSOF.2021.106567.

12 T. Colcombet, N. Fijalkow, and P. Ohlmann. Controlling a random population. *Logical Methods in Computer Science*, Volume 17, Issue 4, November 2021.

13 N. Fijalkow and F. Horn. The surprizing complexity of reachability games. *CoRR*, abs/1010.2420, 2010. arXiv:1010.2420.

14 H. Gimbert, C. Mascle, and P. Totzke. Optimally controlling a random population, 2024. arXiv:2411.15181.

15 D. Harel and A. Pnueli. On the development of reactive systems. In K. Apt, editor, *Logics and Models of Concurrent Systems*, volume F-13 of *NATO Advanced Summer Institutes*, pages 477–498. Springer, 1985.

16 P. Hunter and A. Dawar. Complexity bounds for regular games. In *30th Int. Symp. on Mathematical Foundations of Computer Science*, volume 3618, pages 495–506. Springer, 2005. doi:10.1007/11549345_43.

17 O. Keidar and N. Agmon. Safe navigation in adversarial environments. *Annals of Mathematics and Artificial Intelligence*, 83(2):121–164, 2018. doi:10.1007/S10472-018-9591-0.

18 O. Kupferman, N. Piterman, and M.Y. Vardi. Safraless compositional synthesis. In *Proc. 18th Int. Conf. on Computer Aided Verification*, volume 4144 of *Lecture Notes in Computer Science*, pages 31–44. Springer, 2006. doi:10.1007/11817963_6.

19 O. Kupferman and N. Shenwald. Games with weighted multiple objectives. In *22nd Int. Symp. on Automated Technology for Verification and Analysis*, Lecture Notes in Computer Science. Springer, 2024.

20 E. Lin, N. Agmon, and S. Kraus. Multi-robot adversarial patrolling: Handling sequential attacks. *Artificial Intelligence*, 274, 2019.

21 D.A. Martin. Borel determinacy. *Annals of Mathematics*, 65:363–371, 1975.

22 N. Piterman, A. Pnueli, and Y. Saar. Synthesis of reactive(1) designs. In *Proc. 7th Int. Conf. on Verification, Model Checking, and Abstract Interpretation*, volume 3855 of *Lecture Notes in Computer Science*, pages 364–380. Springer, 2006. doi:10.1007/11609773_24.

23 A. Pnueli and R. Rosner. On the synthesis of a reactive module. In *Proc. 16th ACM Symp. on Principles of Programming Languages*, pages 179–190, 1989.

24 K.S. Senthilkumar and K.K. Bharadwaj. Multi-robot exploration and terrain coverage in an unknown environment. *Robotics and Autonomous Systems*, 60(1):123–132, 2012. doi:10.1016/J.ROBOT.2011.09.005.

25 C.S. Tan, R. Mohd-Mokhtar, and M.R. Arshad. A comprehensive review of coverage path planning in robotics using classical and heuristic algorithms. *IEEE Access*, 9:119310–119342, 2021. doi:10.1109/ACCESS.2021.3108177.

26 E.I. Tolstaya, J. Paulos, V. Kumar, and A. Ribeiro. Multi-robot coverage and exploration using spatial graph neural networks. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8944–8950, 2021.

27 M.Y. Vardi and P. Wolper. Automata-theoretic techniques for modal logics of programs. *Journal of Computer and Systems Science*, 32(2):182–221, 1986. doi:10.1016/0022-0000(86)90026-7.

28 M.Y. Vardi and P. Wolper. Reasoning about infinite computations. *Information and Computation*, 115(1):1–37, 1994. doi:10.1006/INCO.1994.1092.

29 Y. Velner, K. Chatterjee, L. Doyen, T.A. Henzinger, A. Rabinovich, and J-F. Raskin. The complexity of multi-mean-payoff and multi-energy games. *Information and Computation*, 241:177–196, 2015. doi:10.1016/J.IC.2015.03.001.

30 W. Zielonka. Infinite games on finitely coloured graphs with applications to automata on infinite trees. *Theoretical Computer Science*, 200(1-2):135–183, 1998. doi:10.1016/S0304-3975(98)00009-7.

A Missing proofs

A.1 Missing details in the proof of Theorem 2

We start with the case $\gamma = \mathrm{B}$. There, $\mathcal{G} = \langle G', k, E\rangle$, where the game graph $G' = \langle V', \emptyset, v_0, E'\rangle$ has the following components.

1. $V' = V \cup \{v_0\}$, where $v_0 \notin V$.
2. The set of edges E' contains the following edges.
   a. $\langle v_0, v\rangle$, for every $v \in V$. That is, from the initial vertex, Coverer chooses a vertex $v \in V$ by proceeding to the vertex that correspond to v in V'.
   b. $\langle v, v\rangle$, for every $v \in V$.

We prove the correctness of the reduction. Assume first that there exists a vertex-cover $U \subseteq V$ of size k in G. Then, the strategy F_1 for Coverer in $\mathcal{G}$ that sends a different agent to every vertex $v \in U$ is a covering strategy. Indeed, since U is a vertex cover, for every edge $e = \{u, v\} \in E$ we have that $u \in U$ or $v \in U$, thus the Büchi objective e is satisfied in the play in $\mathsf{outcome}(F_1)$ that reaches the self-looped sink v or the self-looped sink u.

For the second direction, assume there exists a covering strategy F_1 in $\mathcal{G}$. Let U be the set of k self-looped sinks that the plays in $\mathsf{outcome}(F_1)$ reach. Then, U is a vertex-cover in G of size k. Indeed, for every edge $e = \{u, v\} \in E$, if every play in $\mathsf{outcome}(F_1)$ does not reach u and does not reach v, then there does not exist a play in $\mathsf{outcome}(F_1)$ that satisfies the Büchi objective e, which is not possible since F_1 is a covering strategy.

For $\gamma = \mathrm{C}$, we define the same graph G, with $\beta = \{V \setminus e : e \in E\}$. Since each play ρ is eventually trapped in a self-loop in a vertex in V, the co-Büchi objective $V \setminus e$ is satisfied in ρ iff the Büchi objective e is satisfied in ρ, thus the game coincides with the one for $\gamma = \mathrm{B}$.

A.2 Missing details in the proof of Theorem 5

We first prove that every strategy $F_1 = \langle f_1^1, f_1^2\rangle$ for Coverer is not a covering strategy in $\mathcal{G}$. If $f_1^1(v_0) = f_1^2(v_0) = v_1$, then F_1 satisfies at most α_1 and α_2; If $f_1^1(v_0) = f_1^2(v_0) = v_2$, then F_1 guarantees the satisfaction of the two objectives α_1 and α_3, when Disruptor proceeds from v_2 to u_2, and the two objectives α_2 and α_3, when Disruptor proceeds from v_2 to d_2; Otherwise, without loss of generality, we have that $f_1^1(v_0) = v_1$ and $f_1^2(v_0) = v_2$. If $f_1^1(v_1) = u_1$, then when Disruptor proceeds from v_2 to u_2, only α_1 and α_3 are satisfied, and in a similar way, if $f_1^1(v_1) = d_1$, then when Disruptor proceeds from v_2 to d_2, only α_2 and α_3 are satisfied. Therefore, F_1 is not a covering strategy.

We now prove that every strategy f_2 for Disruptor is not a disrupting strategy. If $f_2(v_2) = u_2$, then when Coverer uses the strategy in which one agent goes to d_1 and one agent goes to v_2, all three objectives are satisfied; in a similar way, if $f_2(v_2) = d_2$, then when Coverer uses the strategy in which one agent goes to u_1 and one agent goes to v_2, all three objectives are satisfied. Therefore, f_2 is not a disrupting strategy.

A.3 Proof of Theorem 6

For the first claim, consider a CG $\mathcal{G} = \langle G, k, \beta\rangle$, and consider $1 < l \leq k$ and a vertex v of G such that β is (k, l)-decomposable in v. We show that β is also $(k, l-1)$-decomposable in v. The claim for all $l' \leq l$ then follows by repeated (possibly zero) applications of this claim. Since β is (k, l)-decomposable in v, there exists a partition $\beta_1, \ldots, \beta_l$ of β and a partition $A_1, \ldots, A_l$ of A to nonempty sets, such that for every $1 \leq i \leq l$, Coverer wins the game $\langle G^v, |A_i|, \beta_i\rangle$. It follows that Coverer also wins the CG $\langle G^v, |A_{l-1} \cup A_l|, \beta_{l-1} \cup \beta_l\rangle$.

Indeed, the strategy in which the agents in A_{l-1} use their winning strategy for β_{l-1} and the agents in A_l use their winning strategy for β_l covers $\beta_{l-1} \cup \beta_l$. Hence, $\beta_1, \ldots, \beta_{l-2}, \beta_{l-1} \cup \beta_l$ and $A_1, \ldots, A_{l-2}, A_{l-1} \cup A_l$ are partitions of β and A, respectively, to $l-1$ nonempty sets, witnessing that β is $(k, l-1)$-decomposable from v.

For the second item, consider $k \geq 2$. We prove the claim for $l = 2$. By the first item, the result then follows for all $l > 1$. First, for $\gamma = B$, consider the γ-CG $\mathcal{G} = \langle G, k, \beta \rangle$, where G is as follows (see Figure 4 for an example for $k = 2$). Starting from the initial vertex v_0, Disruptor chooses between $k+1$ vertices $v_1, \ldots, v_{k+1}$. From v_i, Coverer chooses among k self-looped sinks $s_i^1, \ldots, s_i^k$, for every $i \in [k+1]$. Then, for every $i \in [k+1]$ and $j \in [i-1]$, the sink s_i^j satisfies the objective α_j, and for every $j \in [k+1] \setminus [i]$, the sink s_i^{j-1} satisfies the objective α_j. The sink s_i^1 also satisfies α_i. That is, every objective $\alpha_j \in \beta \setminus \{\alpha_i\}$ is satisfied in a separate sink reachable from v_i. Formally, $\beta = \{\alpha_i : i \in [k+1]\}$, with $\alpha_i = \{s_j^i : i+1 \leq j \leq k+1\} \cup \{s_j^{i-1} : 1 \leq j \leq i-1\} \cup \{s_i^1\}$, for every $i \in [k+1]$.

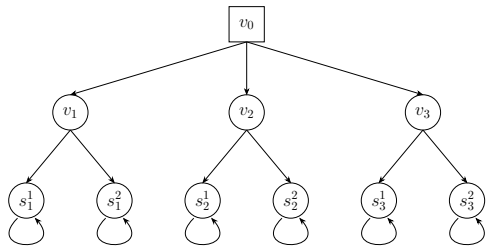

Figure 4 The game graph G for $k = 2$. Here, $\beta = \{\alpha_1, \alpha_2, \alpha_3\}$, with $\alpha_1 = \{s_2^1, s_3^1, s_1^1\}$, $\alpha_2 = \{s_3^2, s_1^1, s_2^1\}$, and $\alpha_3 = \{s_1^2, s_2^2, s_3^1\}$.

It is not hard to see that Coverer wins $\mathcal{G}$. Indeed, β can be satisfied in G^{v_i}, for every successor v_i of v_0. We show that β is not $(k, 2)$-decomposable in G. Assume by contradiction otherwise. Thus, there exist $\beta' \subset \beta$ and $A' \subset A$ such that Coverer wins $\langle G, |A'|, \beta' \rangle$ and $\langle G, |A \setminus A'|, \beta \setminus \beta' \rangle$. Note that every winning strategy for Coverer in $\langle G, |A'|, \beta' \rangle$ requires $|\beta'|$ agents. Indeed, since $|\beta'| \leq k$, Disruptor can choose from the initial vertex a successor v_i such that $\alpha_i \notin \beta'$, thus β' is covered from v_i iff $|\beta'|$ agents are sent to $|\beta'|$ different successors of v_i. In a similar way, every winning strategy for Coverer in $\langle G, |A \setminus A'|, \beta \setminus \beta' \rangle$ requires $|\beta \setminus \beta'|$ agents. Therefore, $k = |A| = |A'| + |A \setminus A'| \geq |\beta'| + |\beta \setminus \beta'| = |\beta| = k+1$, contradiction.

Next, for $\gamma = C$, we define $\mathcal{G}' = \langle G, k, \beta' \rangle$ over the same game graph, now with $\beta' = \{\alpha'_1, \ldots, \alpha'_{k+1}\}$, where $\alpha'_i = \cup(\beta \setminus \{\alpha_i\})$, for every $i \in [k+1]$. Since the vertices s_i^j are sinks, every play in G satisfies the γ objective α'_i iff it satisfies the $\tilde{\gamma}$ objective α_i, for every $i \in [k+1]$, we have that Coverer wins $\mathcal{G}'$ but β' is not $(k, 2)$-decomposable in G.

We continue to the last claim. Consider $k \geq 2$ and $1 \leq l < k$. First, for $\gamma = B$, we describe a γ-CG $\mathcal{G} = \langle G, k, \beta \rangle$ with $\beta = \{\alpha_1, \ldots, \alpha_{k+1}\}$ such that β is (k, l)-decomposable in G, but β is not $(k, l+1)$-decomposable in G. The game is as follows. Let $\mathcal{G}' = \langle G', k-(l-1), \beta' \rangle$ be a CG with $\beta' = \{\alpha'_1, \ldots, \alpha'_{k-(l-1)+1}\}$ such that Coverer wins $\mathcal{G}'$, but β' is not $(k-(l-1), 2)$-decomposable in G'. By item 2, such a CG exists. Starting from the initial vertex in G, Disruptor chooses between l successors $v_1, \ldots, v_l$. The vertices $v_1, \ldots, v_{l-1}$ are self-looped sinks, and from v_l, Coverer proceeds to the sub-graph G'. Then, we define $\alpha_i = \{v_i\}$, for every $i \in [l-1]$, and $\alpha_i = \alpha'_{i-(l-1)}$, for every $l \leq i \leq k+1$. It is not hard to see that β is (k, l)-decomposable in G. Indeed, for every $i \in [l-1]$, Coverer wins $\langle G, 1, \{\alpha_i\} \rangle$, and also Coverer wins $\langle G, k-(l-1), \beta' \rangle$, since Coverer wins $\mathcal{G}'$. We continue to show that β is not $(k, l+1)$-decomposable. Assume by contradiction otherwise, and let $\beta_1, \ldots, \beta_{l+1}$ and $A_1, \ldots, A_{l+1}$ be partitions of β and A to $l+1$ nonempty sets, respectively, such that Coverer

wins $\langle G, |A_i|, \beta_i\rangle$, for every $1 \leq i \leq l+1$. Since there are only $l-1$ objectives in $\beta \setminus \beta'$, we have that β' is divided among at least two different sets in the partition, implying an a-priori partition of β', contradicting the fact β' is not $(k-(l-1), 2)$-decomposable in G'. The same results hold for $\gamma = C$ and the γ-CG $\mathcal{G}' = \langle G, k, \beta''\rangle$ with $\beta'' = \{\alpha''_1, \ldots, \alpha''_{k+1}\}$ such that $\alpha''_i = \cup(\beta \setminus \{\alpha_i\})$, for every $i \in [k+1]$.

A.4 Missing details in the proof of Theorem 11

Formally, $\mathcal{G} = \langle G, |X|, \beta\rangle$, where the game graph $G = \langle V_1, V_2, v_1, E\rangle$ has the following components.

1. $V_1 = \{v_i : i \in [n] \text{ and } Q_i = \exists\} \cup \{l : l \in X \cup \overline{X}\}$. The vertices in $\{v_i : i \in [n] \text{ and } Q_i = \exists\}$ are *existential variable vertices*, and the vertices in $\{l : l \in X \cup \overline{X}\}$ are *literal vertices*.
2. $V_2 = \{v_i : i \in [n] \text{ and } Q_i = \forall\}$. The vertices in $\{v_i : i \in [n] \text{ and } Q_i = \forall\}$ are *universal variable vertices*.
3. The set E of edges includes the following edges.
   a. $\langle v_i, x_i\rangle$ and $\langle v_i, \overline{x_i}\rangle$, for every $i \in [n]$. That is, from the variable vertex v_i, the owner of the vertex chooses an assignment to the variable v_i by proceeding to one of the literal vertices x_i and $\overline{x_i}$, which correspond to assigning **true** and **false**, respectively.
   b. $\langle l, l\rangle$, for every $l \in X \cup \overline{X}$. That is, Coverer can choose to stay in every literal vertex.
   c. $\langle v_i, v_{i+1}\rangle$, for every $i \in [n-1]$ such that $Q_i = \exists$. That is, from existential variable vertices, Coverer can proceed to the next variable vertex.
   d. $\langle l, v_{i+1}\rangle$, for every $i \in [n-1]$ such that $Q_i = \forall$ and $l \in \{x_i, \overline{x_i}\}$. That is, from literal vertices that correspond to universal variables, Coverer can choose between staying in the literal vertex, and proceeding to the next variable vertex.

The set β contains the following objectives.

1. $\alpha_x = \{x, \overline{x}\}$, for every $x \in X$.
2. $\alpha_{C_i} = \{l_i^1, l_i^2, l_i^3\}$, for every $i \in [m]$.

We prove the correctness of the construction. That is, we prove that $\Phi = \textbf{true}$ iff Coverer has a covering strategy in $\mathcal{G}$. Note that the objective α_x, for every $x \in X$, guarantees that exactly one agent has to stay in a vertex that correspond to one of the literals x and $\overline{x}$.

For the first direction, assume $\Phi = \textbf{true}$. Consider the strategy F_1 for Coverer that chooses an assignment to the existential variables, depending on the assignments chosen for preceding variables, in a way that ensures φ is satisfied. Then, for every literal vertex that corresponds to universal variables the play encounters, F_1 leaves one of the agents to traverse its self-loop of indefinitely. Thus, the objective α_x is satisfied in the play that stays in the literal vertex x or $\overline{x}$, for every $x \in X$. And, since the chosen assignment satisfies φ, and literal vertices are visited infinitely often iff they are evaluated to **true** in the chosen assignment, every objective α_{C_i} is satisfied in at least one play. Therefore, F_1 is a covering strategy.

For the second direction, assume there exists a covering strategy F_1 for Coverer in $\mathcal{G}$. Note that by the definition of the construction, F_1 allocates a single agent to stay indefinitely in every literal vertex the play reaches. Also note that for every strategy f_2 for Disruptor, the set of literal vertices that the plays visit infinitely often induce an assignment that satisfy φ. Indeed, otherwise there exists a strategy f_2 for Disruptor that induces an assignment for which there exists a clause C_i all whose literals are evaluated to **false**, thus α_{C_i} is not covered in $\mathsf{outcome}(\langle F_1, f_2\rangle)$, contradicting the fact that F_1 is a covering strategy. Thus, there exists an assignment to the existential variables, depending on assignments to preceding universal variables, that guarantee the sanctification of φ. Accordingly, $\Phi = \textbf{true}$.

A.5 Missing details in the proof of Theorem 14

Consider a strategy $F_1 = \langle f_1^1, \ldots, f_1^k \rangle$ for Coverer. For every $i \in [k]$, let $\rho_i = \mathsf{outcome}(\langle f_1^i, f_2' \rangle)$. Since f_2' is a winning strategy for the ExistsC objective $\{\cup(\beta \setminus \delta)\}_{\delta \in \Delta_{f_2}}$, for every $i \in [k]$ there exist $\delta_i \in \Delta_{f_2}$ such that ρ_i satisfies the co-Büchi objective $\cup(\beta \setminus \delta_i)$, and so the set of Büchi objectives that ρ_i satisfies is a subset of δ_i. Since $\delta_1, \ldots, \delta_k \in \Delta_{f_2}$ and f_2 is a disrupting strategy in $\mathcal{G}$, then by Lemma 13, we have that $\bigcup_{i \in [k]} \delta_i \neq \beta$. Thus, there exists an objective $\alpha_j \in \beta$ such that for every $i \in [k]$, α_j is not satisfied in ρ_i. Accordingly, we have that f_2' is a disrupting strategy for Disruptor in $\mathcal{G}$.

A.6 Missing details in the proof of Theorem 20

First, for the NP part, the algorithm checks that there exists a U-trap strategy for Disruptor that respects g. For that, the algorithm checks that Disruptor can force the play to reach U, checks that Coverer cannot force the play to leave U by verifying that for every vertex $v \in V_1 \cap U$, we have that $\mathsf{succ}(v) \subseteq U$, and then checks that a U-trap strategy can respect g by verifying that for every vertex $v \in V_2 \cap U$, we have that $g(v) \subseteq \mathsf{succ}(v)$.

Then, for the co-NP oracle, the algorithm checks that every U-trap strategy for Disruptor that respects g is a disrupting strategy. For that, the algorithm checks that for every U-trap strategy f_2 for Disruptor that respects g and every strategy $F_1 = \langle f_1^1, \ldots, f_1^k \rangle$ for Coverer, we have that β is not covered in $\mathsf{outcome}(\langle F_1, f_2 \rangle)$. That is, $\bigcap_{i \in [k]} \{\alpha_j \in \beta : \mathit{inf}(\mathsf{outcome}(\langle f_1^i, f_2 \rangle)) \cap \alpha_j \neq \emptyset\} \neq \emptyset$. For that, the algorithm checks that for every k sets $U_1, \ldots, U_k \subseteq U$ of vertices such that for every $i \in [k]$ we have that $U_i = \mathit{inf}(\mathsf{outcome}(\langle f_1, f_2 \rangle))$ for some strategy f_1 for an agent of Coverer and a U-trap strategy f_2 for Disruptor that respects g, we have that $\bigcap_{i \in [k]} \{\alpha_j \in \beta : U_i \cap \alpha_j \neq \emptyset\} \neq \emptyset$. Note that for every set of vertices $U \subseteq V$ and a fairness requirement g for Disruptor, for every U-trap strategy f_2 for Disruptor that respects g and a strategy f_1 for an agent of Coverer, we have that $U' = \mathit{inf}(\mathsf{outcome}(\langle f_1, f_2 \rangle))$ is a strongly connected subset of U such that for every vertex $v \in V_2 \cap U'$ we have that $g(v) \subseteq U'$, and the vertices in $U' \setminus \{g(v) : v \in V_2 \cap U'\}$ are visited infinitely often following choices made by f_1, and thus for every vertex $u \in U' \setminus \{g(v) : v \in V_2 \cap U'\}$ we have that $u \in \{\mathsf{succ}(v) : v \in V_1 \cap U'\}$. Thus, the co-NP oracle checks that for a guessed set of k sets $U_1, \ldots, U_k \subseteq U$ of vertices, for every $i \in [k]$, we have that U_i is strongly connected, $\{g(v) : v \in V_2 \cap U_i\} \subseteq U_i$, and $U_i \setminus \{g(v) : v \in V_2 \cap U_i\} \subseteq \{\mathsf{succ}(v) : v \in V_1 \cap U_i\}$.

A.7 Proof of Theorem 21

We describe reductions from 2QBF. Consider a 2QBF formula $\Phi = \exists X \forall Y \varphi$ with $\varphi = C_1 \vee \cdots \vee C_m$ and $C_i = (l_i^1 \wedge l_i^2 \wedge l_i^3)$, for every $i \in [m]$. For both $\gamma \in \{\text{B,C}\}$, we construct from Φ a γ-CG $\mathcal{G}$ such that $\Phi = \mathbf{true}$ iff Disruptor has a disrupting strategy in $\mathcal{G}$.

We define the γ-CG $\mathcal{G}$ as in the proof of Theorem 11, for the QBF formula $\overline{\Phi} = \forall X \exists Y \overline{\varphi}$. Note that since φ is in 3DNF, the formula $\overline{\varphi}$ is in 3CNF. We prove the correctness of the construction. That is, we prove that $\Phi = \mathbf{true}$ iff Disruptor has a disrupting strategy in $\mathcal{G}$.

For the first direction, assume $\Phi = \mathbf{true}$. Let $\xi : X \to \{\mathbf{true}, \mathbf{false}\}$ be an assignment to the variables in X such that for every assignment $\zeta : Y \to \{\mathbf{true}, \mathbf{false}\}$ to the variables in Y, we have that ξ and ζ satisfy φ. Consider a strategy f_2 for Disruptor that chooses an assignment to the variables in X according to ξ. That is, for every variable $x \in X$, the strategy f_2 proceeds to the literal vertex x if $\xi(x) = \mathbf{true}$, and otherwise proceeds to the literal vertex $\overline{x}$. It is not hard to see that f_2 is a disrupting strategy for Disruptor in

$\mathcal{G}$. Indeed, for every strategy F_1 for Coverer that allocates one agent to stay in the literal vertices of each variable, there exists a clause $\overline{C_i}$ of $\overline{\varphi}$ all whose literals are evaluated to **false** in the corresponding assignment, thus the objective $\alpha_{\overline{C_i}}$ is not covered in $\mathsf{outcome}(\langle F_1, f_2\rangle)$.

For the second direction, assume $\Phi = \textbf{false}$. Then, $\overline{\Phi} = \textbf{true}$, and so Coverer has a covering strategy in $\mathcal{G}$, as shown in the proof of Theorem 11. Consequentially, Disruptor does not have a disrupting strategy in $\mathcal{G}$.