

# Coverage Games<sup>★</sup>

Orna Kupferman<sup>1</sup>[0000–0003–4699–6117] and Noam  
Shenwald<sup>1</sup>[0000–0003–1994–6835]

The Hebrew University of 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 first study the theoretical properties of coverage games, including determinacy, and the ability to a priori decompose the objectives among the agents. We then study the problems of deciding whether the coverer or the disruptor wins. Besides a comprehensive analysis of the tight complexity of the problems, we consider interesting special cases, such as the one-player cases and settings with a fixed number of agents or objectives.

**Keywords:** Two-Player Games,  $\omega$ -Regular Objectives, Coverage, Planning

## 1 Introduction

*Synthesis* is the automated construction of a system from its specification [34]. A *reactive* system interacts with its environment and has to satisfy its specification in all environments [21]. 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

---

<sup>★</sup> This research was supported by the European Research Council, Advanced Grant ADVANSYNT.

token, the generated play satisfies an objective induced by the specification. For example, in *Büchi* games, the objective is given by a set  $\alpha$  of vertices, and a play satisfies the objective if it visits  $\alpha$  infinitely often.

Synthesis is closely related to *automated planning* in AI [37]. In both settings, the goal is to realize a sequence of actions in order to accomplish a desired task, often under uncertainty or adversarial conditions. For example, in *pursuit-evasion* [25], a robot aims to complete a task while avoiding collisions with an adversarial robot. This setting naturally corresponds to a game between the robot and its environment, and extensive work from the synthesis community has been lifted to planning, for both finite- and infinite-horizon tasks [26,18,16,35,15,24,29,5,32,14,20]. The game setting enables the modeling of adversarial or uncertain environments, dealing with unknown terrains, partial observability, or limited sensing capabilities.

Both synthesis and planning often involve *multiple objectives*. For example, in synthesis, generalized objectives such as *generalized Büchi* [7], *generalized parity* [8], *generalized reachability* [17], and *generalized reactivity* (GR(1)) [33,7] specify conjunctions of underlying objectives that the system must satisfy. Research on multi-objective games studies the impact of this conjunction on memory requirements, algorithmic complexity, and strategy construction. Such objectives have also been studied in a variety of game models, including concurrent, stochastic, energy, and weighted games [4,9,6,41,28].

In planning, multiple objectives also naturally arise, especially in multi-agent systems [37]. The agents typically represent robots, drones, or autonomous vehicles that perform tasks such as terrain exploration or handling of requests at various locations [11,37,10,36,38]. For example, in *adversarial patrolling* [30], a team of robots must detect intrusions into a guarded area, and in the *offline-coverage* problem [1], robots must visit every point in a work area — typically within minimal time [9,37,10]. In the context of planning from temporal specifications, researchers have studied settings in which several robots or components act cooperatively toward a shared objective [32,20,16]. For example, in [32], decentralized synthesis methods are used to coordinate a robot swarm so as to satisfy a global LTL specification, assuming full control and a static environment. Similarly, [20] considers multi-robot synthesis from *Signal Temporal Logic* specifications using event-triggered control. Each robot is assigned its own task, and thus there is no strategic cooperation among the robots.

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  $\beta = \{\alpha_1, \dots, \alpha_m\}$  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  $\beta$ . 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  $\beta$ . That is, for all  $1 \leq l \leq m$ , there is an agent  $1 \leq i \leq k$  such that  $\alpha_l$  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  $\beta$  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 in  $\beta$  to a different agent. The interesting cases are when  $1 < k < m$ , in which case the objectives in  $\beta$  need to be partitioned, possibly dynamically, among the agents.

Coverage games significantly extend the settings addressed in current studies of planning. Indeed, in all examples discussed above, either there is a single agents, or the agents are under the full control of the designer (that is, no adversarial environment), or the specifications are pre-assigned to the different agents. Coverage games enable reasoning about multi-agent systems with a shared set of objectives, no a-priori assignment of objectives to agents, and no full control on their behavior.

For example, using coverage games 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 [12].

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  $\beta$ , but rather to prevent Coverer from covering all the objectives in  $\beta$ . Formally, a *disrupting strategy* for Disruptor is a strategy such that for all strategies of Coverer, there is at least one objective in  $\beta$  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  $G$ ,  $\beta$ , 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, thus the coverage and disruption problems do not complement each other.

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 [13,19]). 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 (in particular, when  $k \geq |\beta|$ , the setting corresponds to  $|\beta|$  two-player games with a single objective), 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  $\beta$ , 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 (that is, possibly neither Coverer nor Disruptor has a coverage or disruption strategy), and examine the problem of an *a-priori decomposition* of the objectives in  $\beta$  among the agents. We show that the objectives cannot be decomposed in advance, and characterize covering strategies for Coverer as ones in which all agents satisfy all objectives as long as such a decomposition is impossible.

The characterization is the key to our upper bounds for the complexity of the coverage and disruption problems.

The coverage and disruption problems have three parameters: the game  $G$ , the set  $\beta$  of objectives, and the number  $k$  of agents. We provide a comprehensive *parameterized-complexity* analysis of the problems. Note that fixing  $G$  also fixes  $\beta$ . Also, as cases with  $k \geq |\beta|$  are easy, fixing  $\beta$  essentially fixes  $k$ . Accordingly, we study settings in which only  $k$  or  $|\beta|$  are fixed. Our results are summarized in Table 1. All the complexities in the table are tight.

fixed:	Büchi			co-Büchi		
	–	$k$	$ \beta $	–	$k$	$ \beta $
Coverage	PSPACE Ths. 9, 10	NP Th. 15	PTIME Th. 19	PSPACE Ths. 9, 10	NP Th. 15	PTIME Th. 19
Disruption	$\Sigma_2^P$ Ths. 12, 14	NP Th. 16	PTIME Th. 20	$\Sigma_2^P$ Ths. 13, 14	$\Sigma_2^P$ Th. 17	PTIME Th. 20

**Table 1.** The complexity of the coverage and disruption problems. All complexities are tight.

For both Büchi and co-Büchi objectives, the general problem is PSPACE-complete for coverage and  $\Sigma_2^P$ -complete for disruption, in contrast to the PTIME complexity of standard two-player games with AllB (and the dual ExistsC) and AllC (and the dual ExistsB) objectives. For coverage, fixing the number  $k$  of agents reduces the complexity to NP, and fixing the number  $|\beta|$  of objectives results in PTIME complexity. Essentially, a fixed  $k$  bounds the recursion depth of the general PSPACE algorithm (Theorem 9) by a constant, whereas a fixed  $|\beta|$  enables the elimination of the nondeterminism in the algorithm. Note that for the coverage problem, the complexities for games with Büchi and co-Büchi objectives coincide. For the disruption problem, the picture is different. While the joint complexity as well as the complexity for a fixed  $|\beta|$  coincide for Büchi and co-Büchi, fixing  $k$  only leaves the complexity  $\Sigma_2^P$  for co-Büchi objectives and takes it down to NP in the Büchi case. This is due to the NP-hardness of one-player coverage game with underlying co-Büchi objectives, which already holds for a fixed number of agents.

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 8 and 9), as well as the ability to work with a short symbolic representation of strategies that are not memoryless (Lemma 8). For the lower bounds, the main technical issue is utilizing the agents to model satisfying assignments in Boolean satisfaction problems, and the most technically-involved results are for the cases with a fixed number of agents (Theorems 16 and 17). In particular, for co-Büchi objectives, moving from a single agent to two agents increases the complexity from PTIME to  $\Sigma_2^P$ .

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*  $\rho = v_0, v_1, \dots$  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  $f_i : V^* \cdot V_i \rightarrow V$  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  $\langle \rho, f_i(\rho \cdot v) \rangle \in E$ . 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  $f_i : V_i \rightarrow V$ .

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  $\pi$ . Formally,  $\text{outcome}(\pi) = v_0, v_1, \dots \in V^\omega$  is such that for all  $j \geq 0$ , we have that  $v_{j+1} = f_i(v_0, v_1, \dots, v_j)$ , 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  $\psi$  is an objective for Player 1, specifying a subset of  $V^\omega$ , 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  $\psi$ . 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  $\text{outcome}(\langle f_1, f_2 \rangle)$  satisfies  $\psi$ . 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  $\rho = v_0, v_1, \dots$ , we denote by  $\text{inf}(\rho)$  the set of vertices that are visited infinitely often along  $\rho$ . That is,  $\text{inf}(\rho) = \{v \in V : \text{there are infinitely many } i \geq 0 \text{ such that } v_i = v\}$ . For a set of vertices  $\alpha \subseteq V$ , a play  $\rho$  satisfies the *Büchi* objective  $\alpha$  iff  $\text{inf}(\rho) \cap \alpha \neq \emptyset$ . The objective dual to Büchi is *co-Büchi*. Formally, a play  $\rho$  satisfies a co-Büchi objective  $\alpha$  iff  $\text{inf}(\rho) \cap \alpha = \emptyset$ . We use  $\gamma \in \{B, C\}$  to denote the different objective types.

For a play  $\rho$  and a set of objectives  $\beta = \{\alpha_1, \dots, \alpha_m\}$ , we denote by  $\text{sat}(\rho, \beta)$  the set of objectives  $\alpha_i \in \beta$  that are satisfied in  $\rho$ . An *All objective* is a set  $\beta$  of objectives, all of the same type. A play  $\rho$  satisfies an All objective  $\beta$  iff  $\rho$  satisfies every objective in  $\beta$ . That is, if  $\text{sat}(\rho, \beta) = \beta$ . The objective dual to All is *Exists*.<sup>1</sup> Formally, a play  $\rho$  satisfies an Exists objective  $\beta$  iff  $\rho$  satisfies at least one objective in  $\beta$ . That is, if  $\text{sat}(\rho, \beta) \cap \beta \neq \emptyset$ . Note that an All-co-Büchi objective  $\beta$  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  $\gamma \in \{B, C\}$ , a *two-player multi-agent  $\gamma$ -coverage game* ( $\gamma$ -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  $\beta = \{\alpha_1, \dots, \alpha_m\}$  is a set of objectives of type  $\gamma$ . When  $\gamma$  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  $[k] = \{1, \dots, k\}$ . A *strategy* for Coverer is a tuple  $F_1 = \langle f_1^1, \dots, f_1^k \rangle$  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  $F_1 = \langle f_1^1, \dots, f_1^k \rangle$  for Coverer and  $f_2$  for Disruptor. The outcome of  $\pi$  is the tuple of the  $k$  plays generated when Coverer's agents and Disruptor follow their strategies. Formally,  $\text{outcome}(\pi) = \langle \rho_1, \dots, \rho_k \rangle$ , where  $\rho_i = \text{outcome}(\langle f_1^i, f_2 \rangle)$ , for every  $i \in [k]$ .

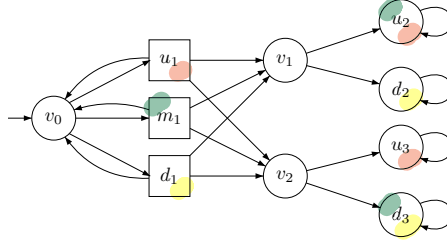
We say that a profile  $\pi$  *covers*  $\beta$  iff every objective in  $\beta$  is satisfied in at least one play in  $\text{outcome}(\pi)$ . That is, if  $\cup\{\text{sat}(\rho, \beta) : \rho \in \text{outcome}(\pi)\} = \beta$ . Equivalently, for every  $j \in [m]$ , there exists  $i \in [k]$  such that  $\alpha_j$  is satisfied in  $\text{outcome}(\langle f_1^i, f_2 \rangle)$ . 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  $\beta$ . 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  $\langle F_1, f_2 \rangle$

<sup>1</sup> All-Büchi objectives are traditionally referred to as *generalized Büchi*. Then, their dual Exists co-Büchi objectives are referred to as *generalized co-Büchi* [27]. We opt to follow a terminology with All and Exists, making the quantification within the satisfaction requirement clearer.

does not cover  $\beta$ . Then, we say that Disruptor wins  $\mathcal{G}$  iff she has a disrupting strategy in  $\mathcal{G}$ . Finally, we say that a  $\gamma$ -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.



**Fig. 1.** The Büchi CG  $\mathcal{G}$ .

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,  $\beta$  is covered. If Disruptor decides to leave the left sub-game and directs the tokens to  $v_1$  or  $v_2$ , the agents split  $\beta$  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,  $\beta$  is covered in this case too.  $\square$

### 3 Special Cases of Coverage Games

As a starter, in this section we examine two special cases of CGs: games in which the number of agents enables an easy reduction to traditional two-player games, and one-player CGs, thus when one player owns all vertices.

#### 3.1 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  $\beta$ . Indeed, the single play induced by the strategy of the single agent has to cover all the objectives in  $\beta$ . 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  $\beta$  to each of the agents. Formally, we have the following.

**Lemma 1.** *Consider a  $\gamma$ -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  $\gamma$ -game  $\mathcal{G}_i = \langle G, \alpha_i \rangle$ .*
- (C3) *There is  $\alpha_i \in \beta$  such that Player 2 wins the two-player  $\gamma$ -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  $\gamma$ -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  $F_1 = \langle f_1^1, \dots, f_1^{|\beta|}, f_1^{|\beta|+1}, \dots, f_1^k \rangle$  is a covering strategy for Coverer, for arbitrary strategies  $f_1^{|\beta|+1}, \dots, f_1^k$  for agents  $|\beta| + 1, \dots, k$ . 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  $\gamma$ -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  $\alpha_i \notin \text{sat}(\rho, \beta)$  for every strategy  $F_1$  for Coverer and every play  $\rho \in \text{outcome}(\langle F_1, f_2 \rangle)$ . Thus, Disruptor has a disrupting strategy and (C4) holds.

By Lemma 1, 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 [31,39,42]

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

### 3.2 One-player CGs

We proceed to 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.

We study the complexity of the coverage and disruption problems in one-player CGs. Our results are summarized in Table 2.

	Büchi			co-Büchi		
fixed:	–	$k$	$ \beta $	–	$k$	$ \beta $
$V = V1$	NP Th. 1	NLOGSPACE Th. 2	NLOGSPACE Th. 4	NP Th. 1	NP Th. 3	NLOGSPACE Th. 4
$V = V2$	NLOGSPACE Th. 5					

**Table 2.** The complexity of deciding one-player CGs. All complexities are tight.

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  $\beta$  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 1.** *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  $\gamma \in \{B, C\}$ , if there exists a path in  $G$  that satisfies a subset  $\beta' \subseteq \beta$  of  $\gamma$  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  $\beta'$ . 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  $\gamma \in \{B, C\}$ , we construct a  $\gamma$ -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  $\beta$  then require these  $k$  vertices to be a vertex cover. For  $\gamma = B$ , 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  $\gamma = C$ , we define  $\beta = \{V \setminus e : e \in E\}$ . Since each play  $\rho$  is eventually trapped in a self-loop in a vertex in  $V$ , the obtained game coincides with the one for  $\gamma = B$ .

We describe the reduction formally and prove their correctness. We start with the case  $\gamma = B$ . 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.  $V' = V \cup \{v_0\}$ , for  $v_0 \notin V$ .
2. The set  $E'$  of edges contains, for every  $v \in V$ , the edges  $\langle v_0, v \rangle$  and  $\langle v, v \rangle$ .  
That is, from the initial vertex, Coverer chooses a vertex  $v \in V$  by proceeding to the vertex that correspond to  $v$  in  $V'$ , and then the play loops forever 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  $\text{outcome}(F_1)$  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  $\text{outcome}(F_1)$  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  $\text{outcome}(F_1)$  does not reach  $u$  and does not reach  $v$ , then there does not exist a play in  $\text{outcome}(F_1)$  that satisfies the Büchi objective  $e$ , which is not possible since  $F_1$  is a covering strategy.

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

When Coverer owns all the vertices and the number of agents is fixed, the problem is easier for  $\gamma = B$ , but retains its NP-hardness for  $\gamma = C$ . Intuitively, the difference in the complexities has to do with the difficulty of finding the maximal sets of objectives that one agent can satisfy on her own. In Büchi CGs, finding these maximal sets is easy, and thus when the number of agents is fixed, finding a combination of them that covers  $\beta$  can be done in NLOGSPACE. On the other hand, in co-Büchi CGs, the hardness of the problem stems from the hardness of finding those maximal sets, and so the problem is NP-hard already for a fixed number of agents. Formally, we have the following.

**Theorem 2.** *The coverage problem for Büchi CGs with  $V = V_1$  and a fixed number of agents is NLOGSPACE-complete.*

*Proof.* Consider a CG  $\langle G, k, \beta \rangle$  with  $V = V_1$ . We first argue that Coverer has a covering strategy iff  $G$  has  $k$  reachable, non-trivial, and strongly-connected components (SCCs)  $C_1, \dots, C_k$  such that  $\bigcup_{i \in [k]} \{\alpha_j \in \beta : \alpha_j \cap C_i \neq \emptyset\} = \beta$ . Indeed, each outcome of the game can reach and stay in an SCC  $C$  of  $G$ , where it can traverse infinitely often all the vertices in  $C$ . Since  $k$  is fixed, an NLOGSPACE algorithm can thus guess  $k$  non-trivial SCCs, and check that every objective  $\alpha \in \beta$  intersects with one of the SCCs.

In more details, the NLOGSPACE algorithm guesses  $k$  vertices  $v_1, \dots, v_k \in V$ , check that they are all reachable from  $v_0$ , and writes them on the tape. Since  $k$  is fixed, the space required is logarithmic in  $|V|$ . Then, for each objective  $\alpha \in \beta$ , the algorithm guesses  $1 \leq i \leq k$  and checks whether there exists a path

from  $v_i$  to  $\alpha$  and back to  $v_i$ , which can be done in NLOGSPACE. If such a path does not exist, the algorithm rejects. If the algorithm successfully completes the check for all the objectives in  $\beta$  it accepts.

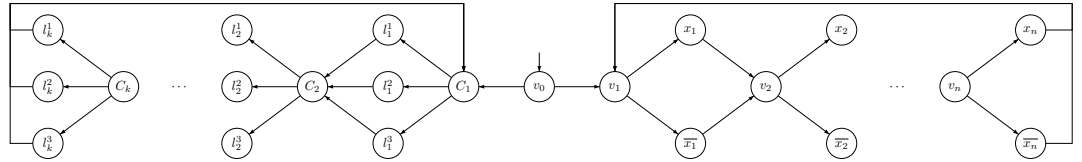
Hardness in NLOGSPACE follows from an easy reduction from deciding one-player Büchi games, a.k.a. nonemptiness of Büchi word automata [40].

While fixing the number of agents reduces the complexity of the coverage problem for  $\gamma = B$ , it does not have the same effect for  $\gamma = C$ . This marks the first difference between the two settings. Formally, we have the following.

**Theorem 3.** *The coverage problem for co-Büchi CGs with  $V = V_1$  is NP-hard, already for CGs with two agents.*

*Proof.* We describe a reduction from 3SAT. Consider a set of variables  $X = \{x_1, \dots, x_n\}$ , and let  $\bar{X} = \{\bar{x}_1, \dots, \bar{x}_n\}$ . Also, let  $\varphi = C_1 \wedge \dots \wedge C_m$  be a propositional Boolean formula given in 3CNF. That is,  $C_i = (l_i^1 \vee l_i^2 \vee l_i^3)$  with  $l_i^1, l_i^2, l_i^3 \in X \cup \bar{X}$ , for every  $i \in [m]$ . We construct from  $\varphi$  a co-Büchi CG  $\mathcal{G} = \langle G, 2, \beta \rangle$ , such that  $V = V_1$  and Coverer has a covering strategy in  $\mathcal{G}$  iff  $\varphi$  is satisfiable.

The game graph  $G$  is defined as follows (see Figure 2). From the initial vertex, Coverer chooses between proceeding to an *assignment sub-graph*, and to a *clause sub-graph*. In the assignment sub-graph, Coverer chooses an assignment to the variables in  $X$ , where choosing an assignment to a variable  $x$  involves choosing between a vertex that corresponds to the literal  $x$ , and a vertex that corresponds to the literal  $\bar{x}$ . Coverer then repeats this process infinitely often.



**Fig. 2.** The game graph  $G$ .

In the clause sub-graph, Coverer chooses a literal for every clause of  $\varphi$ , which involves choosing between vertices that correspond to the literals  $l_i^1, l_i^2$  and  $l_i^3$ , for every  $i \in [m]$ . As in the assignment sub-graph, Coverer then repeats this process.

Coverer has to cover the following objectives. The first two objectives  $\alpha_1$  and  $\alpha_2$  are the vertices of the assignment sub-graph and the vertices of the clause sub-graph, respectively. Note that in order to satisfy both objectives, Coverer must send one agent to the assignment sub-graph, and one agent to the clause

sub-graph. We call those agents the *assignment agent* and the *clause agent*, respectively. Then, for every literal  $l \in X \cup \bar{X}$ , we define the objective  $\alpha_l$  as the set that consists of the vertex that corresponds to  $l$  in the assignment sub-graph, and every vertex that corresponds to  $\bar{l}$  in the clause sub-graph. Accordingly, the objective  $\alpha_l$  is satisfied iff the assignment agent visits the vertex that corresponds to  $l$  only finitely often, or the clause agent visits vertices that correspond to  $\bar{l}$  only finitely often.

Intuitively, if the assignment agent chooses a satisfying assignment and the clause agent chooses literals that are evaluated to **true** in the chosen assignment, then every objective  $\alpha_l$  is satisfied. Hence, a satisfying assignment to  $\varphi$  induces a covering strategy for Coverer. Also, if  $\varphi$  is not satisfiable, then for every assignment induced by the assignment agent, there is a clause in  $\varphi$  that is not satisfied. Then, the clause agent causes the objective of at least one of the literals of this clause to be not satisfied.

Formally, we define  $\mathcal{G} = \langle G, 2, \beta \rangle$  as follows. First, the game graph  $G = \langle V_1, \emptyset, v_0, E \rangle$  contains the following components. (see Fig 2).

1.  $V_1 = \{v_0\} \cup \{v_i : i \in [n]\} \cup X \cup \bar{X} \cup \{C_i : i \in [m]\} \cup \{l_i^1, l_i^2, l_i^3 : i \in [m]\}$ . The vertices in  $\{v_i : i \in [n]\}$  are *variable vertices*, and the vertices in  $\{C_i : i \in [m]\}$  are *clause vertices*. Also, the vertices in  $X \cup \bar{X}$  are *literal vertices*, and the vertices in  $\{l_i^1, l_i^2, l_i^3 : i \in [m]\}$  are *clause-literal vertices*.
2. The set of edges  $E$  contains the following edges.
  - (a)  $\langle v_0, v_1 \rangle$ .
  - (b)  $\langle v_i, x_i \rangle$  and  $\langle v_i, \bar{x}_i \rangle$ , for every  $i \in [n]$ .
  - (c)  $\langle x_i, v_{(i+1) \bmod n} \rangle$  and  $\langle \bar{x}_i, v_{(i+1) \bmod n} \rangle$ , for every  $i \in [n]$ .
  - (d)  $\langle v_0, C_1 \rangle$ .
  - (e)  $\langle C_i, l_i^j \rangle$ , for every  $i \in [m]$  and  $j \in \{1, 2, 3\}$ .
  - (f)  $\langle l_i^j, C_{(i+1) \bmod m} \rangle$ , for every  $i \in [m]$  and  $j \in \{1, 2, 3\}$ .

Then, the set of objectives is  $\beta = \{\alpha_1, \alpha_2\} \cup \{\alpha_l : l \in X \cup \bar{X}\}$ , where  $\alpha_1 = X \cup \bar{X}$ ,  $\alpha_2 = \{l_i^j : i \in [m], j \in \{1, 2, 3\}\}$ , and  $\alpha_l = \{l\} \cup \{\bar{l}_i^j : i \in [m], j \in \{1, 2, 3\}, \text{ and } l_i^j = \bar{l}\}$ , for every  $l \in X \cup \bar{X}$ .

We prove the correctness of the construction. Note that when Coverer allocates both agents to the assignment sub-graph or both agents to the clause sub-graph, one of the objectives  $\alpha_1$  and  $\alpha_2$  is not satisfied in both plays, thus we only consider strategies in which Coverer allocates one agent (the assignment agent) to the assignment sub-graph, and one agent (the clause agent) to the clause sub-graph.

For the first direction, assume  $\varphi$  is satisfiable. Let  $\xi : X \rightarrow \{\mathbf{true}, \mathbf{false}\}$  be a satisfying assignment, and let  $F_1 = \langle f_1^1, f_1^2 \rangle$  be the strategy for Coverer such that  $f_1^1$  is a strategy for the assignment agent that chooses from every variable vertex  $v_i$  the literal vertex  $x_i$  iff  $\xi(x_i) = \mathbf{true}$ , and the literal vertex  $\bar{x}_i$  iff  $\xi(x_i) = \mathbf{false}$ ; The strategy  $f_1^2$  is a strategy for the clause agent that chooses from every clause vertex  $C_i$  a clause-literal vertex  $l_i^j$  such that  $l_i^j$  is evaluated to **true** in  $\xi$ . It is easy to see that  $F_1$  is a covering strategy. Indeed, for every literal  $l \in X \cup \bar{X}$ , if  $l$  is evaluated to **false** in  $\xi$ , then the play in the assignment sub-graph does not

visit the literal vertex  $l$ , thus  $\alpha_l$  is satisfied in the play; and if  $l$  is evaluated to **true**, then the play in the clause sub-graph does not visit clause-literal vertices that correspond to  $\bar{l}$ , thus  $\alpha_l$  is satisfied in the play.

For the second direction, assume  $\varphi$  is not satisfiable. Consider a strategy  $F_1 = \langle f_1^1, f_1^2 \rangle$  for Coverer, and assume WLOG that  $f_1^1$  is for the assignment agent and  $f_1^2$  is for the clause agent. Let  $\xi : X \rightarrow \{\mathbf{true}, \mathbf{false}\}$  be an assignment to the variables in  $X$  such that the assignment agent visits every literal vertex  $l$  with  $\xi(l) = \mathbf{true}$  infinitely often. It is easy to see that there exists a literal  $l \in X \cup \bar{X}$  with  $\xi(l) = \mathbf{true}$  such that the clause agent visits infinitely often clause-literal vertices that corresponds to  $\bar{l}$ . Indeed, otherwise  $\xi$  satisfies  $\varphi$ . Thus,  $\alpha_l$  is not satisfied in both plays, and so  $F_1$  is not a covering strategy.

When  $|\beta|$  is fixed, the problem becomes easier also for  $\gamma = C$ . Intuitively, it follows from the fact that now, there is a fixed number of combinations of  $k$  sets of objectives, one for each agent to satisfy on her own. Thus, finding a combination of them that cover  $\beta$  can be done in NLOGSPACE. Formally, we have the following.

**Theorem 4.** *The coverage problem for Büchi or co-Büchi CGs with  $V = V_1$  and fixed  $|\beta|$  is NLOGSPACE-complete.*

*Proof.* The results for Büchi follow from Theorem 2. We continue to co-Büchi and start with the upper bound. Consider a co-Büchi CG  $\mathcal{G} = \langle G, k, \beta \rangle$  with  $V = V_1$  and fixed  $|\beta|$ . An NLOGSPACE algorithm guesses a partition  $\beta_1, \dots, \beta_k$  of  $\beta$  and writes it on the tape. Note that since  $|\beta|$  is fixed, writing the partition can be done in logarithmic space. Then, for every  $i \in [k]$ , the algorithm checks that there exists a lasso path that satisfies the co-Büchi objective  $\cup \beta_i$ , which can be done in NLOGSPACE.

Hardness in NLOGSPACE follows from an easy reduction from nonemptiness of co-Büchi word automata [40].

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 has the flavor of checking the checking the non-emptiness of nondeterministic automata with acceptance conditions that dualize generalized Büchi or co-Büchi conditions, and is easy.

**Theorem 5.** *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  $\beta$ , namely a path that does not satisfy one of the objectives in  $\beta$ . We argue that the latter can be done in NLOGSPACE, for all  $\gamma \in \{B, C\}$ . Indeed, an NLOGSPACE algorithm can guess a set  $\alpha \in \beta$  and then search for a lasso-shaped path that does not satisfy  $\alpha$ . In case  $\gamma = B$ , the vertices of the lasso loop should be disjoint from  $\alpha$ . In case

$\gamma = C$ , 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 [40].

## 4 Properties of Coverage Games

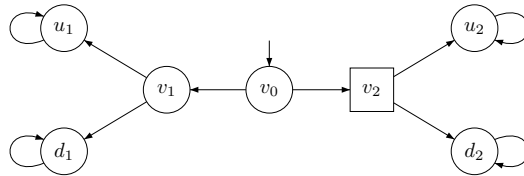
In this section we study theoretical properties of CGs. We start with determinacy and show that CGs need not be determined. We then define and study the *decomposability* of objectives in CGs, namely the ability to a-priori partition the objectives among the agents. We argue that decomposability plays 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 3.1, 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 6.** *For all  $\gamma \in \{B, C\}$ , we have that  $\gamma$ -CGs need not be determined. Moreover, undeterminacy holds already for a  $\gamma$ -CG with only two agents and three objectives.*

*Proof.* For all  $\gamma \in \{B, C\}$ , we describe a  $\gamma$ -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 3.



**Fig. 3.** An undetermined coverage game.

The objectives  $\alpha_1, \alpha_2$ , and  $\alpha_3$  depend on  $\gamma$ . We start with  $\gamma = B$ . There, we define  $\alpha_1 = \{u_1, u_2\}$ ,  $\alpha_2 = \{d_1, d_2\}$ , and  $\alpha_3 = \{u_2, d_2\}$ . Intuitively Coverer has no covering strategy as  $\alpha_3$  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  $\alpha_1$  and  $\alpha_2$  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.

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  $\alpha_1$  and  $\alpha_2$ ; If  $f_1^1(v_0) = f_1^2(v_0) = v_2$ , then  $F_1$  guarantees the satisfaction of the two objectives  $\alpha_1$  and  $\alpha_3$ , when Disruptor proceeds from  $v_2$  to  $u_2$ , and the two objectives  $\alpha_2$  and  $\alpha_3$ , 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  $\alpha_1$  and  $\alpha_3$  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  $\alpha_2$  and  $\alpha_3$  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.

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  $\beta$  among her  $k$  agents. In this section we show that in general, Coverer cannot *a-priori* partition  $\beta$  among the  $k$  agents. Moreover, decompositions of  $\beta$  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  $\beta = \{\alpha_1, \dots, \alpha_m\}$ . Let  $A = [k]$ . For  $1 \leq l \leq k$  and a vertex  $v \in V_1$ , we say that  $\beta$  is  $(k, l)$ -decomposable in  $v$  if Coverer can decompose the task of covering  $\beta$  in  $G^v$  by the agents in  $A$  to  $l$  sub-tasks, each assigned to a different subset of  $A$ . Formally,  $\beta$  is  $(k, l)$ -decomposable in  $v$  if there exists a partition  $\beta_1, \dots, \beta_l$  of  $\beta$  and a partition  $A_1, \dots, A_l$  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  $\beta_i$ . When  $v = v_0$ , we say that  $\beta$  is  $(k, l)$ -decomposable in  $G$ .

We start with some easy observations. First, note that, by definition,  $\beta$  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  $\beta$ , and the definitions coincide. Then,  $\beta$  is  $(k, k)$ -decomposable in  $G$  iff there exists a partition of  $\beta$  to  $k$  sets  $\beta_1, \dots, \beta_k$  such that for every  $1 \leq i \leq k$ , Coverer has a strategy that satisfies the All objective  $\beta_i$ . 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  $\beta_i$ . Finally, the bigger  $l$  is, the more refined is the partition of  $\beta$  to which Coverer commits, making her task harder. Formally, we have the following (see Example 1 for a CG with no a-priori decomposition):

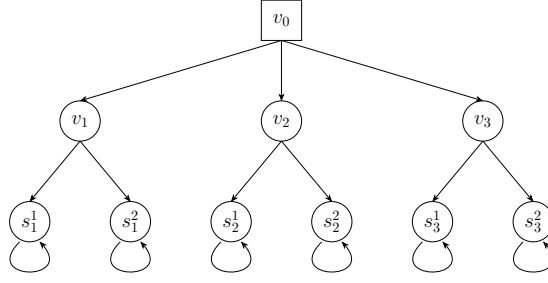
**Theorem 7.** *For all  $\gamma \in \{B, C\}$ , we have the following.*

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

*Proof.* 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  $\beta$  is  $(k, l)$ -decomposable in  $v$ . We show that  $\beta$  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  $\beta$  is  $(k, l)$ -decomposable in  $v$ , there exists a partition  $\beta_1, \dots, \beta_l$  of  $\beta$  and a partition  $A_1, \dots, A_l$  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  $\beta_{l-1}$  and the agents in  $A_l$  use their winning strategy for  $\beta_l$  covers  $\beta_{l-1} \cup \beta_l$ . Hence,  $\beta_1, \dots, \beta_{l-2}, \beta_{l-1} \cup \beta_l$  and  $A_1, \dots, A_{l-2}, A_{l-1} \cup A_l$  are partitions of  $\beta$  and  $A$ , respectively, to  $l - 1$  nonempty sets, witnessing that  $\beta$  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  $\gamma$ -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  $v_1, \dots, v_{k+1}$ . From  $v_i$ , Coverer chooses among  $k$  self-looped sinks  $s_i^1, \dots, s_i^k$ , 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  $\alpha_j$ , and for every  $j \in [k + 1] \setminus [i]$ , the sink  $s_i^{j-1}$  satisfies the objective  $\alpha_j$ . The sink  $s_i^1$  also satisfies  $\alpha_i$ . 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  $i \in [k + 1]$ .

It is not hard to see that Coverer wins  $\mathcal{G}$ . Indeed,  $\beta$  can be satisfied in  $G^{v_i}$ , for every successor  $v_i$  of  $v_0$ . We show that  $\beta$  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  $\beta'$  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



**Fig. 4.** The game graph  $G$  for  $k = 2$ . Here,  $\beta = \{\alpha_1, \alpha_2, \alpha_3\}$ , with  $\alpha_1 = \{s_1^1, s_1^2, s_1^3\}$ ,  $\alpha_2 = \{s_2^1, s_2^2, s_2^3\}$ , and  $\alpha_3 = \{s_3^1, s_3^2, s_3^3\}$ .

$|\beta \setminus \beta'|$  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  $\beta' = \{\alpha'_1, \dots, \alpha'_{k+1}\}$ , 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  $\gamma$  objective  $\alpha'_i$  iff it satisfies the  $\tilde{\gamma}$  objective  $\alpha_i$ , for every  $i \in [k + 1]$ , we have that Coverer wins  $\mathcal{G}'$  but  $\beta'$  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  $\gamma$ -CG  $\mathcal{G} = \langle G, k, \beta \rangle$  with  $\beta = \{\alpha_1, \dots, \alpha_{k+1}\}$  such that  $\beta$  is  $(k, l)$ -decomposable in  $G$ , but  $\beta$  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  $\beta' = \{\alpha'_1, \dots, \alpha'_{k - (l - 1) + 1}\}$  such that Coverer wins  $\mathcal{G}'$ , but  $\beta'$  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  $v_1, \dots, v_l$ . The vertices  $v_1, \dots, v_{l-1}$  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  $\beta$  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  $\beta$  is not  $(k, l + 1)$ -decomposable. Assume by contradiction otherwise, and let  $\beta_1, \dots, \beta_{l+1}$  and  $A_1, \dots, A_{l+1}$  be partitions of  $\beta$  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  $\beta'$  is divided among at least two different sets in the partition, implying an a-priori partition of  $\beta'$ , contradicting the fact  $\beta'$  is not  $(k - (l - 1), 2)$ -decomposable in  $G'$ . The same results hold for  $\gamma = C$  and the  $\gamma$ -CG  $\mathcal{G}' = \langle G, k, \beta'' \rangle$  with  $\beta'' = \{\alpha''_1, \dots, \alpha''_{k+1}\}$  such that  $\alpha''_i = \cup(\beta \setminus \{\alpha_i\})$ , for every  $i \in [k + 1]$ .

Consider a vertex  $v \in V_1$ . For  $l \in [k]$ , we say that  $v$  is a  $(k, l)$ -fork for  $\beta$  if Coverer has a covering strategy  $F_1 = \langle f_1^1, \dots, f_1^k \rangle$  in  $G^v$  such that different agents are sent to  $l$  different successors of  $v$ . Thus, there exist  $l$  successors  $v_1, \dots, v_l$  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 2.** *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  $\beta$ , then  $\beta$  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  $\beta$ . Let  $F_1 = \langle f_1^1, \dots, f_1^k \rangle$  be a winning strategy for Coverer that sends the agents to  $l$  different successors  $v_1, \dots, v_l$  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  $\beta_i$  be the set objectives of  $\beta$  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  $\text{outcome}(F_1^i, f_2)$  that satisfies  $\alpha_j$ . Equivalently,  $\beta_i \subseteq \beta$  is the maximal subset of  $\beta$  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  $\alpha_j$  is not covered in  $\text{outcome}(\langle F_1^i, f_2^i \rangle)$ . 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  $\alpha_j$  is not covered in  $\text{outcome}(\langle F_1, f_2 \rangle)$ , 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  $\beta$ , for some  $2 \leq l \leq k$ . We denote by  $F \subseteq V_1$  the set of vertices that are forks. Let  $V_{\text{avoid}} \subseteq V$  be the set of vertices from which Disruptor can avoid reaching forks. That is,  $v \in V_{\text{avoid}}$  if there exists a strategy  $f_2$  for Player 2 from  $v$  such that for every strategy  $f_1$  for Player 1, the play  $\text{outcome}(\langle f_1, f_2 \rangle)$  does not reach  $F$ . Finally, let  $G_{\text{avoid}} = \langle V'_1, V'_2, v_0, E' \rangle$  be the sub-graph of  $G$  with vertices in  $V_{\text{avoid}}$ . That is,  $V'_1 = V_1 \cap V_{\text{avoid}}$ ,  $V'_2 = V_2 \cap V_{\text{avoid}}$ , and  $E' = E \cap (V_{\text{avoid}} \times V_{\text{avoid}})$ .

*Example 2.* 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  $\beta$  in  $G$  are  $v_1$  and  $v_2$ . Thus, a winning strategy for Coverer cannot a-priori decompose the objectives in  $\beta$ . Note that this is the case also for the covering strategy described in Example 1.

Indeed, in  $v_1$ , Coverer can assign both  $\alpha_1$  and  $\alpha_2$  to an agent that proceeds to  $u_2$  and assign  $\alpha_3$  to an agent that proceeds to  $d_2$ . Similarly, in  $v_2$ , Coverer can assign  $\alpha_1$  to an agent that proceeds to  $u_3$  and assign both  $\alpha_2$  and  $\alpha_3$  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  $\beta$ . Hence,  $v_0$  is not a fork. Finally, as  $\beta$  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  $G_{\text{avoid}}$  is the sub-graph of  $G$  whose vertices are  $v_0, u_1, m_1$  and  $d_1$ .

**Theorem 8.** *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  $v \notin V_{\text{avoid}}$ , or  $v \in V_{\text{avoid}}$  and Coverer wins  $\langle G_{\text{avoid}}^v, 1, \beta \rangle$ .*

*Proof.* We first prove that for all vertices  $v \notin V_{\text{avoid}}$ , Coverer has a strategy to cover  $\beta$  from  $v$ . Consider a vertex  $v \notin V_{\text{avoid}}$ . By the definition of  $V_{\text{avoid}}$ , Coverer has a strategy to reach a fork from  $v$ . Hence, Coverer can cover  $\beta$  from  $v$  by letting all agents follow the same strategy until some  $(k, l)$ -fork  $u$  for  $\beta$  is reached. By Lemma 2,  $\beta$  is  $(k, l)$ -decomposable in  $u$ . Hence, once the  $k$  tokens of the agents reach  $u$ , Coverer can cover  $\beta$  by decomposing it. Let  $F_1^v$  denote a covering strategy for Coverer from a vertex  $v \notin V_{\text{avoid}}$ .

We argue that for every vertex  $v \in V_{\text{avoid}}$ , Coverer has a strategy to cover  $\beta$  from  $v$  iff Coverer wins  $\langle G_{\text{avoid}}^v, 1, \beta \rangle$ . Consider a vertex  $v \in V_{\text{avoid}}$ , and assume first that Coverer has a covering strategy from  $v$ . Since Disruptor can force the play to stay in  $G_{\text{avoid}}$ , every covering strategy for Coverer from  $v$  in  $G$  is also a covering strategy from  $v$  in  $G_{\text{avoid}}$ . Also, for every covering strategy  $F_1 = \langle f_1^1, \dots, f_1^k \rangle$  from  $v$ , the strategies for the different agents agree with each other as long as the play stays in  $G_{\text{avoid}}$ . That is,  $f_1^1(h) = \dots = f_1^k(h)$ , for every  $h \in V_{\text{avoid}}^* \cdot V_1'$ . 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  $V_{\text{avoid}}$ , the sub-graph  $G_{\text{avoid}}$  does not contain forks. Hence, the strategy  $f_1 : V_{\text{avoid}}^* \cdot V_1' \rightarrow V_{\text{avoid}}$  with  $f_1(h) = f_1^1(h)$  for every  $h \in V_{\text{avoid}}^* \cdot V_1'$  is a covering strategy for Coverer  $\langle G_{\text{avoid}}^v, 1, \beta \rangle$ .

For the second direction, assume that Coverer wins  $\langle G_{\text{avoid}}^v, 1, \beta \rangle$ . We show that Coverer has a covering strategy from  $v$ . Let  $f_1$  be a covering strategy for Coverer in  $\langle G_{\text{avoid}}^v, 1, \beta \rangle$ . Consider the strategy  $F_1 = \langle f_1^1, \dots, f_1^k \rangle$  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  $G_{\text{avoid}}$ , 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  $G_{\text{avoid}}$  and reaches a vertex  $u \notin V_{\text{avoid}}$ . For every strategy  $f_2$  for Disruptor, either all the outcomes of  $F_1$  and  $f_2$  stay in  $G_{\text{avoid}}$ , in which case  $\beta$  is covered by each of the agents, or all the outcomes of  $F_1$  and  $f_2$  leave  $G_{\text{avoid}}$ , in which case they follow strategies that cover  $\beta$  by decomposing it.

## 5 The Complexity of the Coverage Problem

In this section, we study the complexity of the coverage problem for Büchi and co-Büchi CGs, and show that it is PSPACE-complete. We study the joint complexity, namely in terms of  $G, \beta$ , and  $k$ . In Sections 7 and 8, we complete the picture with a parameterized-complexity analysis.

We start with upper bounds. Consider a CG  $\mathcal{G} = \langle G, k, \beta \rangle$ . Recall the set  $F$  of forks for  $\beta$ , and the game graph  $G_{\text{avoid}}$ , in which Disruptor can avoid reaching forks. By Theorem 8, Coverer has a covering strategy in  $G$  iff Player 1 has a winning strategy in  $G_{\text{avoid}}$  for the All objective  $\beta$ , and when the play leaves  $G_{\text{avoid}}$  (or if the initial vertex is not in  $G_{\text{avoid}}$ ), Coverer can cover  $\beta$  by reaching a fork, from which  $\beta$  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  $\beta$  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  $\text{NPSpace} = \text{PSPACE}$ , we have the following.<sup>2</sup>

**Theorem 9.** *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  $\beta_1, \dots, \beta_l$  of  $\beta$  and a partition  $A_1, \dots, A_l$  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- $\gamma$  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 8, Coverer has a covering strategy from a vertex  $v$  iff  $v \notin V_{\text{avoid}}$ , or  $v \in V_{\text{avoid}}$  and Coverer wins  $\langle G_{\text{avoid}}^v, 1, \beta \rangle$ . So, if Coverer has a covering strategy, the NTM can guess the set  $F$  of forks for the current set  $\beta$  of objectives and number  $k$  of agents. Then,  $G'$  coincides with  $G_{\text{avoid}}$ , and by Theorem 8, 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 8, it is enough to show that if  $v \in G_{\text{avoid}}$ , then Coverer wins  $\langle G_{\text{avoid}}^v, 1, \beta \rangle$ . Since the vertices in  $U$  are forks,  $G_{\text{avoid}}$  is a sub-graph of  $G'$ . So, since Disruptor can force the play to stay in  $G_{\text{avoid}}$  if  $v$  is in it, we also have that Coverer wins  $\langle G_{\text{avoid}}^v, 1, \beta \rangle$ .

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  $G'$  can be done in polynomial time, and checking whether Player 1 wins an All- $\gamma$  game can be done in polynomial time as well, for both  $\gamma \in \{\text{B}, \text{C}\}$  [39]. Thus, the NTM runs in polynomial space.

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  $G_{\text{avoid}}$ , a covering strategy in  $\langle G_{\text{avoid}}, 1, \beta \rangle$ , and the strategies for reaching forks. Then, for each fork, the strategy describes the partitions of  $\beta$  and  $[k]$ , and the (recursive) description of the strategies in the decomposed game. Note that the description may require more than polynomial space.

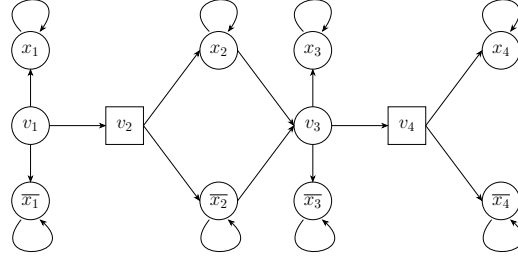
We continue to the lower bounds.

<sup>2</sup> It is not hard to see that the proof of Theorem 9 applies to all prefix-independent objective types  $\gamma$  such that All- $\gamma$  games can be decided in PSPACE.

**Theorem 10.** *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  $X = \{x_1, \dots, x_n\}$  and a QBF formula  $\Phi = Q_1x_1, \dots, Q_nx_n\varphi$  where  $\varphi = C_1 \wedge \dots \wedge C_m$ , 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  $\Phi = \mathbf{true}$  iff Coverer has a covering strategy in  $\mathcal{G}$ .

The game graph  $G$  (see example in Figure 5) lets the players choose an assignment to the variables in  $X$ , following the quantification order of  $\Phi$  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  $\bar{x}_i$ , 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  $\bar{x}_i$ . In those vertices, Coverer chooses between staying in the current vertex, and proceeding to the next variables.



**Fig. 5.** 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  $\alpha_x$  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  $\alpha_{C_i}$  that consists of vertices that correspond to  $C_i$ 's literals. Thus, the joint assignment satisfies  $\varphi$  iff Coverer can cover all the objectives.

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.  $V_1 = \{v_i : i \in [n] \text{ and } Q_i = \exists\} \cup \{l : l \in X \cup \bar{X}\}$ . The vertices in  $\{v_i : i \in [n] \text{ and } Q_i = \exists\}$  are *existential variable vertices*, and the vertices in  $\{l : l \in X \cup \bar{X}\}$  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  $\langle v_i, \bar{x}_i \rangle$ , 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  $\bar{x}_i$ , which correspond to assigning **true** and **false**, respectively.
  - (b)  $\langle l, l \rangle$ , for every  $l \in X \cup \bar{X}$ . 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  $l \in \{x_i, \bar{x}_i\}$ . 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  $\beta$  contains the following objectives.

1.  $\alpha_x = \{x, \bar{x}\}$ , 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  $\Phi = \mathbf{true}$  iff Coverer has a covering strategy in  $\mathcal{G}$ . Note that the objective  $\alpha_x$ , 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  $\bar{x}$ .

For the first direction, assume  $\Phi = \mathbf{true}$ . 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  $\varphi$  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  $\alpha_x$  is satisfied in the play that stays in the literal vertex  $x$  or  $\bar{x}$ , for every  $x \in X$ . And, since the chosen assignment satisfies  $\varphi$ , and literal vertices are visited infinitely often iff they are evaluated to **true** in the chosen assignment, every objective  $\alpha_{C_i}$  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  $\varphi$ . 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  $\alpha_{C_i}$  is not covered in  $\text{outcome}(\langle F_1, f_2 \rangle)$ , 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  $\varphi$ . Accordingly,  $\Phi = \mathbf{true}$ .

For co-Büchi CGs, note that, as in previous reductions, a Büchi objective  $\alpha \subseteq (X \cup \bar{X})$  is satisfied in a play  $\rho$  iff the co-Büchi objective  $(X \cup \bar{X}) \setminus \alpha$  is satisfied in  $\rho$ , and thus the same reduction with dual objectives applies. Hence, Coverer has a covering strategy in the Büchi CG  $\mathcal{G} = \langle G, |X|, \beta \rangle$  iff Coverer has a covering strategy in the co-Büchi CG  $\mathcal{G}' = \langle G, |X|, \beta' \rangle$ , with  $\beta' = \{(X \cup \bar{X}) \setminus \alpha : \alpha \in \beta\}$ . Therefore,  $\Phi = \mathbf{true}$  iff Coverer has a covering strategy in  $\mathcal{G}'$ .

## 6 The Complexity of the Disruption Problem

In this section, we study the complexity of the disruption problem for 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  $\Sigma_2^P$ -complete. Thus, at least in terms of its theoretical complexity class, it is easier than the coverage problem. In Sections 7 and 7, we complete the picture with a parameterized-complexity analysis of the 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  $\delta$  when Player 2 uses  $f_2$ . That is,  $\Delta_{f_2}$  is the set of maximal sets  $\delta \subseteq \beta$  for which there exists a strategy  $f_1$  for Player 1 such that  $\text{sat}(\text{outcome}(\langle f_1, f_2 \rangle), \beta) = \delta$ .

**Lemma 3.** *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  $\text{outcome}(\langle f_1, f_2 \rangle)$  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  $\rho = \text{outcome}(\langle f_1, f_2 \rangle)$ . By the definition of  $\Delta_{f_2}$ , there exists  $\delta \in \Delta_{f_2}$  such that  $\text{sat}(\rho, \beta) \subseteq \delta$ . Accordingly,  $(\beta \setminus \delta) \cap \text{sat}(\rho, \beta) = \emptyset$ , and so  $\rho$  satisfies all the  $\tilde{\gamma}$  objectives in  $\beta \setminus \delta$ .

Back to CGs, the intuition behind the definition of  $\Delta_{f_2}$  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  $\delta$ . Formally, we have the following, which follows directly from the definitions.

**Lemma 4.** *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  $\delta_1, \dots, \delta_k \in \Delta_{f_2}$ , we have that  $\bigcup_{i \in [k]} \delta_i \neq \beta$ .*



*Proof.* A strategy  $f_2$  is a disrupting strategy iff for every strategy  $F_1$  for Coverer, there exists an objective  $\alpha_i \in \beta$  such that  $\alpha_i$  is not satisfied in  $\rho$ , for every play  $\rho \in \text{outcome}(\langle F_1, f_2 \rangle)$ . Since the sets in  $\Delta_{f_2}$  are the maximal sets of objectives that Coverer can satisfy while Disruptor uses  $f_2$ , the above claim follows.

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

**Theorem 11.** *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 3, for every strategy  $f_1$  for an agent of Coverer, there exists  $\delta \in \Delta_{f_2}$  such that  $\text{outcome}(\langle f_1, f_2 \rangle)$  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  $f'_2$  exists. We prove that  $f'_2$  is a disrupting strategy for Disruptor in  $\mathcal{G}$ . Consider a strategy  $F_1 = \langle f_1^1, \dots, f_1^k \rangle$  for Coverer. For every  $i \in [k]$ , let  $\rho_i = \text{outcome}(\langle f_1^i, f'_2 \rangle)$ . 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  $\rho_i$  satisfies the co-Büchi objective  $\cup(\beta \setminus \delta_i)$ , and so the set of Büchi objectives that  $\rho_i$  satisfies is a subset of  $\delta_i$ . Since  $\delta_1, \dots, \delta_k \in \Delta_{f_2}$  and  $f_2$  is a disrupting strategy in  $\mathcal{G}$ , then by Lemma 4, we have that  $\bigcup_{i \in [k]} \delta_i \neq \beta$ . Thus, there exists an objective  $\alpha_j \in \beta$  such that for every  $i \in [k]$ ,  $\alpha_j$  is not satisfied in  $\rho_i$ . Accordingly, we have that  $f'_2$  is a disrupting strategy for Disruptor in  $\mathcal{G}$ .

The fact we can restrict attention to memoryless disrupting strategies leads to an upper bound for the disruption problem, which we show to be tight.

**Theorem 12.** *The disruption problem for Büchi CGs is in  $\Sigma_2^P$ .*

*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 1, deciding the existence of covering strategies in  $G_{f_2}$  can be done in NP, implying the  $\Sigma_2^P$ -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  $\rho$  iff at least one AllB objective in  $\mathcal{F}$  is satisfied in  $\rho$  [22]. By [22], 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 [22].

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

**Lemma 5.** *Consider an AllB game  $\mathcal{G} = \langle G, \delta \rangle$ , with  $\delta = \{\alpha_1, \dots, \alpha_m\}$ . 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  $\delta$ , and that  $\{g_1^i : i \in [m]\}$  are its *witnesses*.

Lemma 6 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  $\inf(\text{outcome}(\langle f_1, f_2 \rangle)) \subseteq U$ , for every strategy  $f_2$  for Player 2.

A *fairness requirement* for Player 1 is a function  $g : V_1 \rightarrow 2^V$  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  $v \in \inf(\text{outcome}(\langle f_1, f_2 \rangle))$ , then  $g(v) \subseteq \inf(\text{outcome}(\langle f_1, f_2 \rangle))$ .

**Lemma 6.** *Consider an AllB game  $\mathcal{G} = \langle G, \delta \rangle$ , with  $\delta = \{\alpha_1, \dots, \alpha_m\}$ . 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  $\alpha_i$ , where it switches to follow  $g_1^{(i+1) \bmod m}$ . 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  $\delta$  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  $U = \inf(\text{outcome}(\langle f_1, f_2 \rangle))$  does not intersect with all the sets in  $\delta$ . 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  $U' = \inf(\text{outcome}(\langle f_1^*, f'_2 \rangle))$  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  $\text{outcome}(\langle f_1^*, f'_2 \rangle)$  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  $\text{outcome}(\langle f_1, f_2 \rangle)$ , thus all the vertices in  $g(v)$  are visited infinitely often.

By Lemma 6, 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  $\Delta$  is not empty. That is,  $\bigcap_{i \in [k]} \delta_i \neq \emptyset$ , for every  $\delta_1, \dots, \delta_k \in \Delta$ .

**Lemma 7.** *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  $\delta_1, \dots, \delta_k \in \Delta_{f_2}$ . By the definition of  $\Delta_{f_2}$ , there exist strategies  $f_1^1, \dots, f_1^k$  for the agents of Coverer with  $\text{sat}(\text{outcome}(\langle f_1^i, f_2 \rangle), \beta) = \delta_i$ , for every  $i \in [k]$ . Since  $f_2$  is a disrupting strategy, then, by Lemma 4, 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  $F_1 = \langle f_1^1, \dots, f_1^k \rangle$  for Coverer and  $\delta_1, \dots, \delta_k \in \Delta_{f_2}$  with  $\text{sat}(\text{outcome}(f_1^i, f_2)) \subseteq \delta_i$ , for every  $i \in [k]$ , there exists  $\alpha_j \in \bigcap_{i \in [k]} (\beta \setminus \delta_i)$ . Thus,  $\alpha_j$  is not covered in  $\text{outcome}(\langle F_1, f_2 \rangle)$ . Accordingly,  $f_2$  is a disrupting strategy.

**Lemma 8.** *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  $\delta_{f_2}$  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  $\delta_{f_2}$ , then, by Lemma 6, 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  $\delta_{f_2}$ , 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 7, 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  $\delta_1, \dots, \delta_k \in \Delta_{f_2}$  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 13.** *The disruption problem for co-Büchi CGs is in  $\Sigma_2^P$ .*

*Proof.* An NP algorithm that uses a co-NP oracle guesses a set of vertices  $U \subseteq V$  and a fairness requirement  $g : (V_2 \cap U) \rightarrow (2^U \setminus \{\emptyset\})$ , 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 8, the search can be restricted to trap strategies that respect a given fairness requirement.

We describe how the above two steps can indeed be done in NP and co-NP. 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  $\text{succ}(v) \subseteq U$ , 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  $g(v) \subseteq \text{succ}(v)$ .

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  $F_1 = \langle f_1^1, \dots, f_1^k \rangle$  for Coverer, we have that  $\beta$  is not covered in  $\text{outcome}(\langle F_1, f_2 \rangle)$ . That is,  $\bigcap_{i \in [k]} \{\alpha_j \in \beta : \inf(\text{outcome}(\langle f_1^i, f_2 \rangle)) \cap \alpha_j \neq \emptyset\} \neq \emptyset$ . For that, the algorithm checks that for every  $k$  sets  $U_1, \dots, U_k \subseteq U$  of vertices such that for every  $i \in [k]$  we have that  $U_i = \inf(\text{outcome}(\langle f_1, f_2 \rangle))$  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  $U' = \inf(\text{outcome}(\langle f_1, f_2 \rangle))$  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  $u \in \{\text{succ}(v) : v \in V_1 \cap U'\}$ . Thus, the co-NP oracle checks that for a guessed set of  $k$  sets  $U_1, \dots, U_k \subseteq U$  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  $U_i \setminus \{g(v) : v \in V_2 \cap U_i\} \subseteq \{\text{succ}(v) : v \in V_1 \cap U_i\}$ .

We continue to lower bounds. Note that the game in the reduction in the proof of Theorem 10 is determined when  $\Phi$  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  $\Phi = \mathbf{false}$ , which holds iff the 2QBF formula  $\exists X \forall Y \bar{\varphi}$  is valid, Disruptor has a disrupting strategy. Since 2QBF is  $\Sigma_2^P$ -hard, we have the following.

**Theorem 14.** *The disruption problem for Büchi or co-Büchi CGs is  $\Sigma_2^P$ -hard.*

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

We define the  $\gamma$ -CG  $\mathcal{G}$  as in the proof of Theorem 10, for the QBF formula  $\bar{\Phi} = \forall X \exists Y \bar{\varphi}$ . Note that since  $\varphi$  is in 3DNF, the formula  $\bar{\varphi}$  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  $\xi : X \rightarrow \{\mathbf{true}, \mathbf{false}\}$  be an assignment to the variables in  $X$  such that for every assignment  $\zeta : Y \rightarrow \{\mathbf{true}, \mathbf{false}\}$  to the variables in  $Y$ , we have that  $\xi$  and  $\zeta$  satisfy  $\varphi$ . Consider a strategy  $f_2$  for Disruptor that chooses an assignment to the variables in  $X$  according to  $\xi$ . 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  $\bar{x}$ . 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  $\bar{C}_i$  of  $\bar{\varphi}$  all whose literals are evaluated to  $\mathbf{false}$  in the corresponding assignment, thus the objective  $\alpha_{\bar{C}_i}$  is not covered in  $\text{outcome}(\langle F_1, f_2 \rangle)$ .

For the second direction, assume  $\Phi = \mathbf{false}$ . Then,  $\bar{\Phi} = \mathbf{true}$ , and so Coverer has a covering strategy in  $\mathcal{G}$ , as shown in the proof of Theorem 10. Consequently, Disruptor does not have a disrupting strategy in  $\mathcal{G}$ .

## 7 CGs with a Fixed Number of Agents

Consider a CG  $\mathcal{G} = \langle G, k, \beta \rangle$ . In typical applications, the game graph  $G$  is much bigger than  $k$  and  $\beta$ . It is thus interesting to examine the complexity of CGs in terms of the size of  $G$ , assuming the other parameters are fixed. In this section we study the complexity of deciding Büchi and co-Büchi CGs when the number  $k$  of agents is fixed.

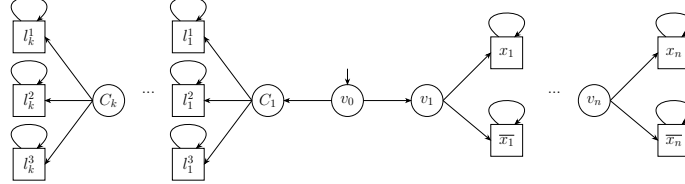
We start with the coverage problem.

**Theorem 15.** *The coverage problem for Büchi or co-Büchi CGs with a fixed number of agents is NP-complete. Hardness in NP holds already for CGs with two agents.*

*Proof.* We start with the upper bound. When the number  $k$  of agents is fixed, the depth of the recursion performed by the NTM described in the proof of Theorem 9 is fixed, and so the NTM runs in polynomial time. Hence, deciding whether Coverer has a covering strategy Büchi or co-Büchi CG with a fixed number of agents can be done in NP.

We continue to the lower bounds. In Theorem 3, we show that deciding whether Coverer has a covering strategy in a co-Büchi CG with  $k = 2$  is NP-hard already for the case  $V = V_1$ .

For Büchi CGs, we describe a reduction from 3SAT. Consider a set of variables  $X = \{x_1, \dots, x_n\}$ , and let  $\varphi = C_1 \wedge \dots \wedge C_m$  with  $C_i = (l_i^1 \vee l_i^2 \vee l_i^3)$  and  $l_i^1, l_i^2, l_i^3 \in X \cup \overline{X}$ , for every  $i \in [m]$ . We construct from  $\varphi$  a CG  $\mathcal{G} = \langle G, 2, \beta \rangle$  such that Coverer has a covering strategy in  $\mathcal{G}$  iff  $\varphi$  is satisfiable.



**Fig. 6.** Coverer has a covering strategy in  $\mathcal{G}$  iff  $\varphi$  is satisfiable.

The game graph  $G$  is as follows (see Fig. 6). From the initial vertex, Coverer chooses between proceeding (right) to an *assignment sub-graph*, and (left) to a *clause sub-graph*. In the assignment sub-graph, Coverer chooses an assignment to the variables in  $X$  by visiting assignment-literal vertices. The assignment-literal vertices are owned by Disruptor, who chooses between staying in the assignment-literal vertex and proceeding to the next variable. In the clause sub-graph, Coverer chooses a literal for every clause of  $\varphi$ . She does so by visiting clause-literal vertices. The clause-literal vertices are owned by Disruptor, who chooses between staying in the clause-literal vertex and proceeding to the next

clause. Note that plays in  $G$  do not return to  $v_0$ , thus each play eventually gets stuck in some literal vertex.

We define the objectives in  $\mathcal{G}$  as follows. The first two objectives  $\alpha_1$  and  $\alpha_2$  are the sets of vertices of the assignment and clause sub-graphs, ensuring that Coverer allocates one agent to each sub-graph. We call them the *assignment agent* and the *clause agent*, respectively. In addition, for every literal  $l \in X \cup \bar{X}$ , we define the objective  $\alpha_l$  as the union of the set of assignment-literal vertices that correspond to literals  $l' \neq l$  with the set of clause-literal vertices that correspond to literals  $l' \neq \bar{l}$ . Note that the objective  $\alpha_l$  is satisfied iff the play in the assignment sub-graph stays in an assignment-literal vertex that does not correspond to  $l$  or the play in the clause sub-graph stays in a clause-literal vertex that does not correspond to  $\bar{l}$ . Accordingly, if the assignment agent chooses a satisfying assignment, and the clause agent chooses literals that are evaluated to **true** according to chosen assignment, then every objective  $\alpha_l$  is satisfied, no matter where Disruptor chooses the plays to stay in the different sub-graphs. On the other hand, if the assignment agent chooses a literal  $l$  and the clause agent chooses the literal  $\bar{l}$ , then Disruptor can choose to stay in those vertices in the respective sub-graphs, in which case  $\alpha_l$  is not satisfied.

Formally,  $\mathcal{G} = \langle G, 2, \beta \rangle$  is defined as follows. First, the game graph  $G = \langle V_1, V_2, v_0, E \rangle$  contains the following components.

1.  $V_1 = \{v_0\} \cup \{v_i : i \in [n]\} \cup \{C_i : i \in [m]\}$ . The vertices in  $\{v_i : i \in [n]\}$  are *variable vertices*, and the vertices in  $\cup\{C_i : i \in [m]\}$  are *clause vertices*.
2.  $V_2 = X \cup \bar{X} \cup \{l_i^1, l_i^2, l_i^3 : i \in [m]\}$ . The vertices in  $X \cup \bar{X}$  are *literal vertices*, and the vertices in  $\{l_i^1, l_i^2, l_i^3 : i \in [m]\}$  are *clause-literal vertices*.
3. The set of edges  $E$  contains the following edges.
  - (a)  $\langle v_0, v_1 \rangle$ .
  - (b)  $\langle v_i, x_i \rangle$  and  $\langle v_i, \bar{x}_i \rangle$ , for every  $i \in [n]$ .
  - (c)  $\langle l, l \rangle$ , for every  $l \in X \cup \bar{X}$ .
  - (d)  $\langle x_i, v_{i+1} \rangle$  and  $\langle \bar{x}_i, v_{i+1} \rangle$ , for every  $i < n$ .
  - (e)  $\langle v_0, C_1 \rangle$ .
  - (f)  $\langle C_i, l_i^j \rangle$ , for every  $i \in [m]$  and  $j \in \{1, 2, 3\}$ .
  - (g)  $\langle l_i^j, l_i^j \rangle$ , for every  $i \in [m]$  and  $j \in \{1, 2, 3\}$ .
  - (h)  $\langle l_i^j, C_{i+1} \rangle$ , for every  $i < m$  and  $j \in \{1, 2, 3\}$ .

The set of objectives is  $\beta = \{\alpha_1, \alpha_2\} \cup \{\alpha_l : l \in X \cup \bar{X}\}$ , where  $\alpha_1 = X \cup \bar{X}$ ,  $\alpha_2 = \{l_i^j : i \in [m], j \in \{1, 2, 3\}\}$ , and  $\alpha_l = ((X \cup \bar{X}) \setminus \{l\}) \cup \{l_i^j : i \in [m], j \in \{1, 2, 3\}, \text{ and } l_i^j \neq \bar{l}\}$ , for every  $l \in X \cup \bar{X}$ .

We prove the correctness of the construction. Note that for two plays  $\rho_1$  and  $\rho_2$  in the assignment and clause sub-graphs, respectively, there are literal and clause literal vertices  $l$  and  $l_i^j$  such that  $\inf(\rho_1) = l$  and  $\inf(\rho_2) = l_i^j$ . Then,  $\beta$  is covered in  $\{\rho_1, \rho_2\}$  iff  $l_i^j \neq \bar{l}$ . Indeed,  $\alpha_{l'}$  is satisfied in  $\rho_1$  for every  $l' \in (X \cup \bar{X}) \setminus \{l\}$ , and  $\alpha_l$  is satisfied in  $\rho_2$  iff  $l_i^j \neq \bar{l}$ .

For the first direction, assume  $\varphi$  is satisfiable. Let  $\xi : X \rightarrow \{\mathbf{true}, \mathbf{false}\}$  be a satisfying assignment, and let  $F_1 = \langle f_1^1, f_1^2 \rangle$  be the strategy for Coverer such that  $f_1^1$  is a strategy for the assignment agent that chooses from every

variable vertex  $v_i$  the literal vertex  $x_i$  iff  $\xi(x_i) = \mathbf{true}$ , and the literal vertex  $\bar{x}_i$  iff  $\xi(x_i) = \mathbf{false}$ ; The strategy  $f_1^2$  for the clause agent chooses from every clause vertex  $C_i$  a clause-literal vertex  $l_i^j$  such that  $l_i^j$  is evaluated to  $\mathbf{true}$  in  $\xi$ . It is easy to see that  $F_1$  is a covering strategy. Indeed, for every strategy for Disruptor, there does not exist a literal  $l$  such that the play in the assignment sub-graph visits the literal vertex  $l$ , and the play in the clause sub-graph visits a clause-literal vertex that corresponds to  $\bar{l}$ .

For the second direction, assume Coverer has a covering strategy  $F_1 = \langle f_1^1, f_1^2 \rangle$ . Assume WLOG that  $f_1^1$  is a strategy for the assignment agent and  $f_1^2$  is a strategy for the clause agent. Let  $\xi$  be the assignment to the variables in  $X$  that agrees with  $f_1^1$ . That is, a literal  $l$  is evaluated to  $\mathbf{true}$  in  $\xi$  iff  $f_1^1$  proceeds from the appropriate variable vertex to the literal vertex  $l$ . It is easy to see that  $\xi$  satisfies  $\varphi$ . Indeed, otherwise there exists a clause  $C_i$  all whose literals are evaluated to  $\mathbf{false}$  in  $\xi$ , thus Disruptor can make the play in the clause sub-graph to stay in the clause-literal vertex  $l_i^j$  the clause agent chooses from  $C_i$ , and make the play in the assignment sub-graph to stay in the literal vertex  $l$  such that  $l_i^j = \bar{l}$ . When Disruptor uses such a strategy, we have that  $\alpha_l$  is not covered, contradicting the fact  $F_1$  is a covering strategy.

We continue to the disruption problem.

**Theorem 16.** *The disruption problem for Büchi CGs with a fixed number of agents is NP-complete. Hardness in NP holds already for two agents.*

*Proof.* We start with the upper bound. Consider a Büchi CG  $\mathcal{G} = \langle G, k, \beta \rangle$ . An NP algorithm 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  $f_2$ . By Theorem 11, it is sufficient to consider memoryless strategies for Disruptor. By Theorem 2, verifying the strategy against a fixed number of agents can be done in NLOGSPACE, and hence also in polynomial time.

For the lower bound, we describe a reduction from 3SAT. Consider a set of variables  $X = \{x_1, \dots, x_n\}$ , and consider  $\varphi = C_1 \wedge \dots \wedge C_m$  with  $C_i = (l_i^1 \vee l_i^2 \vee l_i^3)$  and  $l_i^1, l_i^2, l_i^3 \in X \cup \bar{X}$ , for every  $i \in [m]$ . We construct from  $\varphi$  a CG  $\mathcal{G} = \langle G', 2, \beta \rangle$  such that Disruptor has a disrupting strategy in  $\mathcal{G}$  iff  $\varphi$  is satisfiable.

Recall the game graph  $G$  from the proof of Theorem 15. The game graph  $G'$  is similar to  $G$ , except that the roles of the players in the assignment and clause sub-graphs are dualized. That is, Disruptor owns the variable and clause vertices, and Coverer owns the literal vertices. The vertex  $v_0$  is still owned by Coverer.

We define the objectives in  $\mathcal{G}$  as follows. Similar to the CG in the proof of Theorem 15, we define objectives  $\alpha_1$  and  $\alpha_2$  that force Coverer to allocate one agent to each sub-graph. In addition, for every literal  $l \in X \cup \bar{X}$ , we define the objective  $\alpha_l$  to be the union of the set of assignment-literal vertices that correspond to literals  $l' \neq l$  with the set of clause-literal vertices that correspond to  $\bar{l}$ . Note that the objective  $\alpha_l$  is satisfied iff the play in the assignment sub-graph stays in an assignment literal vertex that does not correspond to  $l$ , or the



play in the clause sub-graph stays in a clause-literal vertex that corresponds to  $\bar{l}$ . Accordingly, if Disruptor chooses a satisfying assignment in the assignment sub-graph, and chooses literals in the clause sub-graph that are evaluated to **true** according to chosen assignment, then there exists an objective  $\alpha_l$  that is not satisfied, no matter where Coverer chooses the plays to get trapped in the different sub-graphs. Also, if Disruptor chooses a literal  $l$  in the assignment sub-graph and the literal  $\bar{l}$  in the clause sub-graph, Coverer can choose to stay in those vertices in the respective sub-graphs, in which case  $\alpha_l$  is satisfied. Thus,  $\varphi$  is satisfiable iff Disruptor has a disrupting strategy.

Formally, the set of objectives is  $\beta = \{\alpha_1, \alpha_2\} \cup \{\alpha_l : l \in X \cup \bar{X}\}$ , where  $\alpha_1 = X \cup \bar{X}$ ,  $\alpha_2 = \{l_i^j : i \in [m], j \in \{1, 2, 3\}\}$ , and  $\alpha_l = ((X \cup \bar{X}) \setminus \{l\}) \cup \{l_i^j : i \in [m], j \in \{1, 2, 3\}, \text{ and } l_i^j = \bar{l}\}$ , for every  $l \in X \cup \bar{X}$ .

We prove the correctness of the construction. Note that for two plays  $\rho_1$  and  $\rho_2$  in the assignment and clause sub-graphs, respectively, there are assignment-literal and clause-literal vertices  $l$  and  $l_i^j$  such that  $\inf(\rho_1) = l$  and  $\inf(\rho_2) = l_i^j$ . Then,  $\beta$  is covered in  $\{\rho_1, \rho_2\}$  iff  $l_i^j = \bar{l}$ . Indeed,  $\alpha_{l'}$  is satisfied in  $\rho_1$  for every  $l' \in (X \cup \bar{X}) \setminus \{l\}$ , and  $\alpha_l$  is satisfied in  $\rho_2$  iff  $l_i^j = \bar{l}$ .

It is easy to see that  $\varphi$  is satisfiable iff Disruptor has a disrupting strategy. Indeed, when Disruptor only chooses literal vertices that correspond to literals evaluated to **true** by a satisfying assignment, we have that  $l_i^j \neq \bar{l}$ , for every literal vertices  $l$  and  $l_i^j$  in the two sub-graphs in which the plays get trapped, thus  $\alpha_l$  is not satisfied. For the second direction, every disrupting strategy for Disruptor chooses an assignment in the assignment sub-graph and clause-literal vertices in the clause sub-graph such that every clause-literal Disruptor chooses corresponds to a literal that is evaluated to **true** by the chosen assignment, as then  $l_i^j \neq \bar{l}$ , for every literal vertex  $l$  and clause-literal vertex  $l_i^j$ . Coverer chooses to stay indefinitely in.

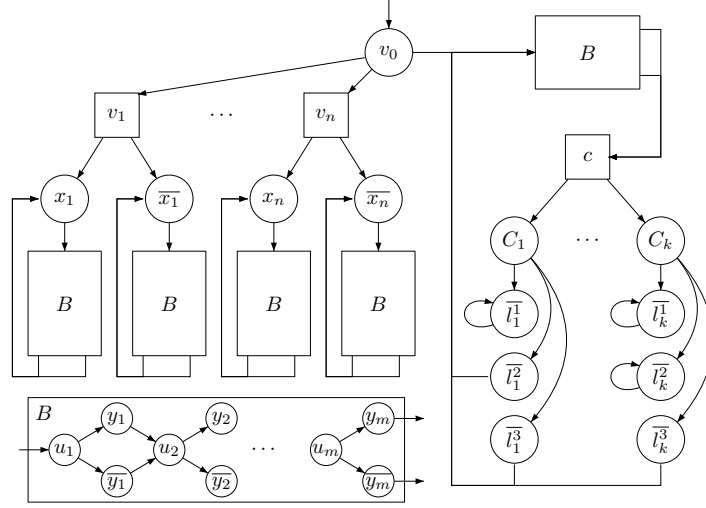
Although fixing the number of agents significantly reduces the complexity of the disruption problem for  $\gamma = B$ , the complexity remains surprisingly unchanged for  $\gamma = C$ , even when there are only two agents. Formally, we have the following.

**Theorem 17.** *The disruption problem for co-Büchi CGs with a fixed number of agents is  $\Sigma_2^P$ -complete. Hardness in  $\Sigma_2^P$  holds already for two agents.*

*Proof.* The  $\Sigma_2^P$  upper bound follows from Theorem 13. For the lower bound, we describe a reduction from 2QBF. Consider sets of variables  $X = \{x_1, \dots, x_n\}$  and  $Y = \{y_1, \dots, y_m\}$ , and let  $\bar{X} = \{\bar{x}_1, \dots, \bar{x}_n\}$  and  $\bar{Y} = \{\bar{y}_1, \dots, \bar{y}_m\}$ . Let  $Z = X \cup Y$  and  $\bar{Z} = \bar{X} \cup \bar{Y}$ . Consider a 2QBF formula  $\Phi = \exists X \forall Y \varphi$  with  $\varphi = C_1 \vee \dots \vee C_k$  and  $C_i = (l_i^1 \wedge l_i^2 \wedge l_i^3)$  and  $l_i^1, l_i^2, l_i^3 \in Z \cup \bar{Z}$ , for every  $i \in [k]$ . We construct from  $\Phi$  a co-Büchi CG  $\mathcal{G} = \langle G, 2, \beta \rangle$  such that  $\Phi = \mathbf{true}$  iff Disruptor has a disrupting strategy in  $\mathcal{G}$ .

The game graph  $G$  (see Fig. 7) contains an *assignment* sub-graph, and a *refute* sub-graph. In the assignment sub-graph, Coverer chooses a variable  $x_i \in X$ , Disruptor chooses an assignment to  $x_i$ , and then Coverer repeatedly visits the

literal vertex that corresponds to the assignment and chooses an assignment to the variables in  $Y$ . Note that while in a given play, Disruptor chooses an assignment to a single variable in  $X$ , a strategy for Disruptor in the assignment sub-graph induces an assignment to all variables in  $X$ . In the refute sub-graph, Coverer tries to refute the assignment Disruptor chooses for the variables in  $X$ . For that, Coverer chooses an assignment to the variables in  $Y$ , then Disruptor chooses a clause  $C_i$  of  $\varphi$ , essentially claiming that all its literals are evaluated to **true** in the joint assignment, and then Coverer chooses a literal  $l_i^j$  of  $C_i$ , essentially claiming that  $l_i^j$  is evaluated to **false** in the joint assignment. If  $l_i^j$  is an  $X$ -literal, the game proceeds to a self-looped sink that corresponds to  $\overline{l_i^j}$ . If  $l_i^j$  is a  $Y$ -literal, the game proceeds to a vertex that corresponds to  $\overline{l_i^j}$ , and returns to the initial vertex in the refute sub-graph.



**Fig. 7.** The game graph  $G$ , when  $l_1^1, l_k^1, l_k^2 \in X \cup \overline{X}$ . The sub-graph  $B$  appears on the bottom (left).

The objectives are defined so that Coverer is forced to allocate one agent to the assignment sub-graph (the assignment agent) and one agent to the refute sub-graph (the refute agent). In addition, for every literal  $l \in Y \cup \overline{Y}$ , we define an objective  $\alpha_l$  that is satisfied in plays that visit finitely often vertices that correspond to  $l$  in both sub-graphs. Finally, for every literal  $l \in X \cup \overline{X}$ , we define an objective  $\alpha_l$  that is satisfied in the refute sub-graph if the play there visits finitely often vertices that correspond to  $l$ , and is satisfied in the assignment sub-graph if the play there visits finitely often  $X$ -literal vertices that do not correspond to  $l$ . Accordingly, if Disruptor chooses an assignment  $\xi$  to the variables in  $X$  such that  $\varphi$  is satisfied for every assignment to the variables in  $Y$ , Disruptor is

able to always choose a clause all whose literals are evaluated to **true** in the joint assignment, and thus force the refute agent to proceed to a vertex that corresponds to a literal  $l$  evaluated to **false** in the joint assignment. If  $l$  is an  $X$ -literal, the objective  $\alpha_l$  cannot be covered, because Disruptor chooses  $\bar{l}$  in the assignment sub-graph. Otherwise, there exists a  $Y$ -literal  $l$  such that the refute agent chooses both vertices that correspond to  $l$  and  $\bar{l}$  infinitely often. Note that then, one of the objectives  $\alpha_l$  and  $\alpha_{\bar{l}}$  cannot be covered, as the assignment agent is forced to visit vertices that correspond to  $l$  or  $\bar{l}$  infinitely often.

On the other hand, if Disruptor chooses an assignment to the variables in  $X$  for which there exists an assignment to the variables in  $Y$  such that the joint assignment does not satisfy  $\varphi$ , then the refute agent can choose this assignment, and thus she can choose for each clause a literal evaluated to **false** in the joint assignment, in which case the refute agent only visits vertices that correspond to literals that are evaluated to **true**. In this case, Coverer has a strategy that, together with the strategy for Disruptor, covers all the objectives.

Formally, we define  $\mathcal{G} = \langle G, 2, \beta \rangle$  as follows. First, the game graph  $G = \langle V_1, V_2, v_0, E \rangle$  contains the following components.

1.  $V_1 = \{v_0\} \cup (X \cup \bar{X}) \cup \{u_i^l, y_i^l, \bar{y}_i^l : l \in X \cup \bar{X}, i \in [m]\} \cup \{u_i : i \in [m]\} \cup (Y \cup \bar{Y}) \cup \{C_i : i \in [k]\} \cup \{\bar{l}_i^j : i \in [k], j \in \{1, 2, 3\}\}$ . The vertices in  $(X \cup \bar{X})$  are *X-literal vertices*, the vertices in  $\{u_i^l : i \in [m]\}$  are *the Y-variable vertices of  $l$*  and the vertices in  $\{y_i^l, \bar{y}_i^l : i \in [m]\}$  are *the Y-literal vertices of  $l$* , for every literal  $l \in (X \cup \bar{X})$ , the vertices in  $\{u_i, i \in [m]\}$  are *Y-variable vertices*, the vertices in  $(Y \cup \bar{Y})$  are *Y-literal vertices*, the vertices in  $\{C_i : i \in [k]\}$  are *clause vertices*, and the vertices in  $\{\bar{l}_i^j : i \in [k], j \in \{1, 2, 3\}\}$  are *refute-literal vertices*.
2.  $V_2 = \{v_i : i \in [n]\} \cup \{c\}$ . The vertices in  $\{v_i : i \in [n]\}$  are *X-variable vertices*, and  $c$  is a *clause-choosing vertex*.
3. The set of edges  $E$  consists of the following edges.
  - (a)  $\langle v_0, v_i \rangle$ , for every  $i \in [n]$ , and  $\langle v_0, u_1 \rangle$ . That is, from the initial vertex, Coverer chooses an  $X$ -variable, or proceed to  $u_1$  which is the initial vertex of the refute sub-graph.
  - (b)  $\langle v_i, x_i \rangle$  and  $\langle v_i, \bar{x}_i \rangle$ , for every  $i \in [n]$ . That is, from the  $X$ -variable vertex  $v_i$ , Disruptor assigns **true** to  $x_i$  by proceeding to the  $X$ -literal vertex  $x_i$ , and assigns **true** to  $x_i$  by proceeding to the  $X$ -literal vertex  $\bar{x}_i$ .
  - (c)  $\langle l, u_1^l \rangle$ , for every  $l \in X \cup \bar{X}$ .
  - (d)  $\langle u_i^l, y_i^l \rangle$  and  $\langle u_i^l, \bar{y}_i^l \rangle$ , for every  $l \in X \cup \bar{X}$  and  $i \in [m]$ .
  - (e)  $\langle y_i^l, u_{i+1}^l \rangle$  and  $\langle \bar{y}_i^l, u_{i+1}^l \rangle$ , for every  $l \in X \cup \bar{X}$  and  $i \leq m - 1$ .
  - (f)  $\langle y_m^l, l \rangle$  and  $\langle \bar{y}_m^l, l \rangle$ , for every  $l \in X \cup \bar{X}$ .
  - (g)  $\langle u_i, y_i \rangle$  and  $\langle u_i, \bar{y}_i \rangle$ , for every  $i \in [m]$ .
  - (h)  $\langle y_i, u_{i+1} \rangle$  and  $\langle \bar{y}_i, u_{i+1} \rangle$ , for every  $i \leq m - 1$ .
  - (i)  $\langle y_m, c \rangle$  and  $\langle \bar{y}_m, c \rangle$ .
  - (j)  $\langle c, C_i \rangle$ , for every  $i \in [k]$ .
  - (k)  $\langle C_i, \bar{l}_i^j \rangle$ , for every  $i \in [k]$  and  $j \in \{1, 2, 3\}$ .
  - (l)  $\langle \bar{l}_i^j, \bar{l}_i^j \rangle$ , for every  $i \in [k]$  and  $j \in \{1, 2, 3\}$  such that  $\bar{l}_i^j \in X \cup \bar{X}$ .

(m)  $\langle \bar{l}_i^j, u_1 \rangle$ , for every  $i \in [k]$  and  $j \in \{1, 2, 3\}$  such that  $l_i^j \in Y \cup \bar{Y}$ .

The set of objectives is  $\beta = \{\alpha_1, \alpha_2\} \cup \{\alpha_l : l \in X \cup \bar{X} \cup Y \cup \bar{Y}\}$ , where

1.  $\alpha_1 = X \cup \bar{X}$ ,
2.  $\alpha_2 = \{\bar{l}_i^j : i \in [k], j \in \{1, 2, 3\}\}$ .
3. For every literal  $l \in Y \cup \bar{Y}$ , we have that  $\alpha_l = \{l\} \cup \{l' : l' \in X \cup \bar{X}\} \cup \{\bar{l}_i^j : i \in [k], j \in \{1, 2, 3\}, \text{ and } \bar{l}_i^j = l\}$ . That is,  $\alpha_l$  consist of all the vertices in both sub-graphs that correspond to the literal  $l$ .
4. For every literal  $l \in X \cup \bar{X}$ , we have that  $\alpha_l = ((X \cup \bar{X}) \setminus \{l\}) \cup \{\bar{l}_i^j : i \in [k], j \in \{1, 2, 3\}, \text{ and } \bar{l}_i^j = l\}$ . That is,  $\alpha_l$  consists of all of the vertices in the refute sub-graph that correspond to  $l$ , and all the  $X$ -literal vertices in the assignment sub-graph that do not correspond to  $l$ .

We prove the correctness of the construction. That is, we prove that  $\Phi = \mathbf{true}$  iff Disruptor has a disrupting strategy in  $\mathcal{G}$ . Note that the objectives  $\alpha_1$  and  $\alpha_2$  force Coverer to allocate one agent to each sub-graph, so we only refer to strategies for Coverer that do so, and we call the respective agents the assignment agent, and the refute agent.

For the first direction, assume  $\Phi = \mathbf{true}$ . Consider a witness assignment  $\xi$  to the variables in  $X$ , and let  $f_2$  be a strategy for Disruptor that chooses  $X$ -literal vertices that correspond to literals evaluated to **true** in  $\xi$  in the assignment sub-graph, and in the refute sub-graph chooses a clause  $C_i$  all whose literals are evaluated to **true** in  $\xi$  and the latest assignment the refute agent chose to the variables in  $Y$ . We show that  $f_2$  is a disrupting strategy. Consider a strategy  $F_1 = \langle f_1^1, f_1^2 \rangle$  for Coverer, assume WLOG that  $f_1^1$  is a strategy for the refute agent and  $f_1^2$  is a strategy for the assignment agent. Also, let  $\rho_1 = \text{outcome}(\langle f_1^1, f_2 \rangle)$  and  $\rho_2 = \text{outcome}(\langle f_1^2, f_2 \rangle)$ . Note that since Disruptor only chooses clauses all whose literals are evaluated to **true** in  $\xi$  and the latest assignment to the variables in  $Y$ , the refute agent chooses a refute-literal vertex  $\bar{l}_i^j$  such that  $\bar{l}_i^j$  is evaluated to **false** in this assignment.

If the play  $\rho_1$  reaches a refute-literal vertex  $\bar{l}_i^j$  such that  $\bar{l}_i^j = l$  for some literal  $l \in X \cup \bar{X}$ , we have that  $\alpha_l$  is not covered in  $\text{outcome}(\langle F_1, f_2 \rangle)$ . Indeed, since  $\bar{l}_i^j \in \alpha_l$ , the objective  $\alpha_l$  is not satisfied in the play  $\rho_1$ . Also, the play in the assignment sub-graph satisfied  $\alpha_l$  iff it visits finitely often  $X$ -literal vertices that correspond to literals  $l' \neq l$ , and thus visits the  $X$ -literal vertex  $l$  infinitely often. However,  $\rho_2$  visits infinitely often an  $X$ -literal vertex that corresponds to a literal that is evaluated to **true** in  $\xi$ , and  $l$  is evaluated to **false** in  $\xi$ .

Otherwise, the play  $\rho_1$  visits finitely often refute-literal vertices that correspond to literals in  $X \cup \bar{X}$ . After each time the refute agent chooses an assignment to the variables in  $Y$ , she is forced to visit a refute-literal vertex  $\bar{l}_i^j$  such that  $\rho_1$  previously visits the  $Y$ -literal vertex that corresponds to  $l_i^j$ . Thus, there exists a literal  $l \in Y \cup \bar{Y}$  such that both objectives  $\alpha_l$  and  $\alpha_{\bar{l}}$  are not satisfied in  $\rho_1$ . Since  $\rho_2$  satisfies at most one objective between  $\alpha_l$  and  $\alpha_{\bar{l}}$ , we have that one of them is not covered in  $\text{outcome}(\langle F_1, f_2 \rangle)$ . Therefore,  $f_2$  is a disrupting strategy.

For the second direction, assume  $\Phi = \mathbf{false}$ . For a strategy  $f_2$  for Disruptor, let  $\xi$  be the assignment that Disruptor chooses to the variables in  $X$  in the assignment sub-graph, and let  $\zeta$  be an assignment to the variables in  $Y$  such that  $\varphi$  is not satisfied by  $\xi$  and  $\zeta$ . We show that  $f_2$  is not a covering strategy.

Consider the strategy  $F_1 = \langle f_1^1, f_1^2 \rangle$  for Coverer such that the strategy  $f_1^1$  for the refute agent only chooses  $Y$ -literal vertices that correspond to literals that are evaluated to **true** in  $\zeta$ , and for each clause  $C_i$ , chooses a refute-literal vertex  $\overline{l_i^j}$  such that  $l_i^j$  is evaluated to **false** in  $\xi$  and  $\zeta$ , and thus  $\overline{l_i^j}$  is evaluated to **true** in  $\xi$  and  $\zeta$ .

Also, the strategy  $f_1^2$  is for the assignment agent. There, it is defined as follows.

If  $\text{outcome}(\langle f_1^1, f_2 \rangle)$  reaches a refute-literal vertex  $\overline{l_i^j}$  such that  $\overline{l_i^j} = l$  for some  $l \in X \cup \overline{X}$ , we define the strategy  $f_1^2$  that chooses the  $X$ -variable vertex  $x_i$  such that  $l \in \{x_i, \overline{x_i}\}$ . Note that since  $l$  is evaluated to **true** in  $\xi$ , the strategy  $f_2$  proceeds from the  $X$ -variable vertex  $x_i$  to the  $X$ -literal vertex  $l$ . It is not hard to see that  $\beta$  is covered in  $\text{outcome}(\langle F_1, f_2 \rangle)$ . Indeed,  $l \in X \cup \overline{X}$ , and so for every literal  $l' \in Y \cup \overline{Y} \cup ((X \cup \overline{X}) \setminus \{l\})$ , we have that  $\alpha_{l'}$  is satisfied in the play in the refute sub-graph, and since the play in the assignment sub-graph visits finitely often  $X$ -literal vertices that correspond to literals in  $((X \cup \overline{X}) \setminus \{l\})$ , we have that  $\alpha_l$  is satisfied in the play in the assignment sub-graph.

Otherwise,  $\text{outcome}(\langle f_1^1, f_2 \rangle)$  avoids refute-literal vertices that correspond to literals in  $X \cup \overline{X}$ . Then, we define the strategy  $f_1^2$  that chooses some  $X$ -variable vertex, and then only chooses  $Y$ -literal vertices of the appropriate literal that correspond to literals evaluated to **false** in  $\zeta$ . Note that then the play in the refute sub-graph visits finitely often vertices that correspond to literals in  $Y \cup \overline{Y}$  that are evaluated to **false** in  $\zeta$ , and the play in the assignment sub-graph visits finitely often vertices that correspond to literals in  $Y \cup \overline{Y}$  that are evaluated to **true** in  $\zeta$ . Also, for every literal  $l \in X \cup \overline{X}$  we have that  $\alpha_l$  is satisfied in the play in the refute sub-graph, as it visits finitely often refute-literal vertices that correspond to  $l$ . Thus,  $\beta$  is covered in  $\text{outcome}(\langle F_1, f_2 \rangle)$ .

Therefore,  $f_2$  is not a disrupting strategy.

## 8 CGs with a Fixed Number of Underlying Objectives

In this section we continue the parameterized-complexity analysis of Büchi and co-Büchi CGs and analyze the complexity of CGs when the number of underlying objectives is fixed. We show that both problems are PTIME-complete, regardless of whether the number of agents is fixed too.

Consider a CG  $\mathcal{G} = \langle G, k, \beta \rangle$ . As detailed in Section 3.1, when  $k \geq |\beta|$ , both the coverage and disruption in  $\mathcal{G}$  correspond to  $|\beta|$  two-player games with a single objective (see Lemma 1). Since two-player Büchi and co-Büchi games are PTIME-complete [39,23], we have the following:

**Theorem 18.** *The coverage and disruption problems for Büchi or co-Büchi CGs with a fixed  $|\beta|$  and  $k \geq |\beta|$  are PTIME-complete.*

Accordingly, for the rest of the section we assume that  $k < |\beta|$ , and so fixing  $|\beta|$  also fixes  $k$ . We start with deciding whether Coverer has a covering strategy.

**Theorem 19.** *The coverage problems for Büchi or co-Büchi CGs with fixed numbers of agents and underlying objectives are PTIME-complete.*

*Proof.* Consider a CG  $\mathcal{G} = \langle G, k, \beta \rangle$ . Recall the upper bound proof of Theorem 9, which describes an NTM that runs in polynomial space. The nondeterminism is used to guess forks, where each guess involves guessing a partition of  $\beta$  among the  $k$  agents and then recursively checking that Coverer has covering strategies in the resulting CGs. When both  $\beta$  and  $k$  are fixed, the number of possible partitions is constant, and the nondeterministic guesses can be replaced by a deterministic enumeration of all options. This yields a DTM that also runs in polynomial space.

Furthermore, as shown in Theorem 15, fixing  $k$  bounds the recursion depth by a constant. Thus, when both  $k$  and  $|\beta|$  are fixed, the entire procedure runs in polynomial time, and the overall complexity is PTIME.

Hardness in PTIME follows from the PTIME-hardness of Büchi and co-Büchi games [23], which correspond to the special case with  $k = |\beta| = 1$ .

We continue to the disruption problem.

**Theorem 20.** *The disruption problems for Büchi or co-Büchi CGs with fixed numbers of agents and underlying objectives are PTIME-complete.*

*Proof.* Consider a CG  $\mathcal{G} = \langle G, k, \beta \rangle$ . Recall that for a strategy  $f_2$  for Disruptor in  $G$ , the set  $\Delta_{f_2} \subseteq 2^\beta$  consists of the maximal subsets  $\beta' \subseteq \beta$  such that there exists a strategy  $f_1$  for Coverer with which  $\text{sat}(\text{outcome}(\langle f_1, f_2 \rangle), \beta) = \beta'$ . Also recall that, by Lemma 4, a strategy  $f_2$  for Disruptor is a disrupting strategy in  $\mathcal{G}$  iff for every  $k$  sets  $\delta_1, \dots, \delta_k \in \Delta_{f_2}$ , we have  $\bigcup_{i \in [k]} \delta_i \neq \beta$ .

Based on this characterization, we can construct polynomial-time algorithms for the disruption problems as follows. The algorithms iterate over sets  $\Delta \subseteq 2^\beta$  such that for all  $\delta_1, \dots, \delta_k \in \Delta$ , the union  $\bigcup_{i \in [k]} \delta_i$  is a strict subset of  $\beta$ , and checks whether there exists a strategy  $f_2$  for Disruptor such that  $\Delta_{f_2} \subseteq \Delta$ .

When the underlying objectives are Büchi, each such set  $\Delta$  induces the ExistsC objective  $\{\cup(\beta \setminus \delta) : \delta \in \Delta\}$ , and the algorithm checks whether Player 2 has a winning strategy for this objective. In the case of co-Büchi objectives, the set  $\Delta$  induces the superset objective  $\{(\beta \setminus \delta) : \delta \in \Delta\}$ , which the algorithm translates to an equivalent AllB objective and checks whether Player 2 has a winning strategy for it.

Since  $\beta$  is fixed, the number of candidate sets  $\Delta$  and the size of the induced objectives are fixed, and so all steps can be performed in polynomial time.

## 9 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 and a single agent to CGs adds the challenge of decomposing the set  $\beta$  of objectives among the agents. For example, while multiple co-Büchi objectives can be merged, making them easier than multiple Büchi objectives, our results show that co-Büchi CGs are not easier (in fact, for the case of a fixed number of agents, even harder) 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  $F_2 = \langle f_2^1, \dots, f_2^l \rangle$  of strategies, and Coverer should cover all the objectives in  $\beta$  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  $\beta$  iff there is an a-priori decomposition of  $\beta$  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.
3. R. Bloem, K. Chatterjee, and B. Jobstmann. Graph games and reactive synthesis. In *Handbook of Model Checking.*, pages 921–962. Springer, 2018.
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.
5. A. Camacho, J. A. Baier, C. Muise, and S. A. McIlraith. Finite LTL synthesis as planning. In *Proceedings of the 28th International Conference on Automated Planning and Scheduling*, volume 28, pages 29–38, 2018.
6. 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.
7. 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.
8. 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.
9. 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.
10. 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.
11. H. Choset. Coverage for robotics - a survey of recent results. *Annals of Mathematics and Artificial Intelligence*, 31:113–126, 2001.
12. 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.
13. T. Colcombet, N. Fijalkow, and P. Ohlmann. Controlling a random population. *Logical Methods in Computer Science*, Volume 17, Issue 4, 11 2021.
14. G. De Giacomo, A. Di Stasio, L. M. Tabajara, M. Y. Vardi, and S. Zhu. Finite?trace and generalized?reactivity specifications in temporal synthesis. *Formal Methods in System Design*, 61(3):485–510, 2021.



15. G. De Giacomo and M.Y. Vardi. Synthesis for LTL and LDL on finite traces. In *Proc. 24th Int'l Joint Conf. on Artificial Intelligence*, pages 1558–1564. AAAI Press, 2015.
16. R. Ehlers, S. Seshia, and H. Kress-Gazit. Synthesis with identifiers. In *Proc. 15th Int. Conf. on Verification, Model Checking, and Abstract Interpretation*, volume 8318 of *Lecture Notes in Computer Science*, pages 415–433. Springer, 2014.
17. N. Fijalkow and F. Horn. The surprising complexity of reachability games. *CoRR*, abs/1010.2420, 2010.
18. C. Finucane, G. Jing, and H. Kress-Gazit. Ltlmop: Experimenting with language, temporal logic and robot control. In *IEEE/RSJ Int'l. Conf. on Intelligent Robots and Systems*, pages 1988 – 1993, 2010.
19. H. Gimbert, C. Mascle, and P. Totzke. Optimally controlling a random population, 2024.
20. D. Gundana and H. Kress-Gazit. Event-based signal temporal logic synthesis for single and multi-robot tasks. *IEEE Robotics Autom. Lett.*, 6(2):3687–3694, 2021.
21. 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.
22. 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.
23. N. Immerman. Number of quantifiers is better than number of tape cells. *Journal of Computer and Systems Science*, 22(3):384–406, 1981.
24. B. Johnson and H. Kress-Gazit. Analyzing and revising synthesized controllers for robots with sensing and actuation errors. *I. J. Robotics Res.*, 34(6):816–832, 2015.
25. O. Keidar and N. Agmon. Safe navigation in adversarial environments. *Annals of Mathematics and Artificial Intelligence*, 83(2):121?164, 2018.
26. H. Kress-Gazit, G. E. Fainekos, and G. J. Pappas. Where's waldo? sensor-based temporal logic motion planning. In *IEEE International Conference on Robotics and Automation*, pages 3116–3121, 2007.
27. O. Kupferman, N. Piterman, and M.Y. Vardi. Safrless compositional synthesis. In *Proc. 18th Int. Conf. on Computer Aided Verification*, volume 4144 of *Lecture Notes in Computer Science*, pages 31–44. Springer, 2006.
28. 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.
29. M. Lahijanian, M.R. Maly, D. Fried, L. E. Kavraki, H. Kress-Gazit, and M.Y. Vardi. Iterative temporal planning in uncertain environments with partial satisfaction guarantees. *IEEE Trans. Robotics*, 32(3):583–599, 2016.
30. E. Lin, N. Agmon, and S. Kraus. Multi-robot adversarial patrolling: Handling sequential attacks. *Artificial Intelligence*, 274, 2019.
31. D.A. Martin. Borel determinacy. *Annals of Mathematics*, 65:363–371, 1975.
32. S. Moarref and H. Kress-Gazit. Automated synthesis of decentralized controllers for robot swarms from high-level temporal logic specifications. *Auton. Robots*, 44(3-4):585–600, 2020.
33. 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.
34. 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.

35. V. Raman, N. Piterman, C. Finucane, and H. Kress-Gazit. Timing semantics for abstraction and execution of synthesized high-level robot control. *IEEE Trans. Robotics*, 31(3):591–604, 2015.
36. 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.
37. 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.
38. E.I. Tolstaya, J. Paulos, V. Kumar, and A. Ribeiro. Multi-robot coverage and exploration using spatial graph neural networks. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 8944–8950, 2021.
39. 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.
40. M.Y. Vardi and P. Wolper. Reasoning about infinite computations. *Information and Computation*, 115(1):1–37, 1994.
41. 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.
42. W. Zielonka. Infinite games on finitely coloured graphs with applications to automata on infinite trees. *Theoretical Computer Science*, 200(1-2):135–183, 1998.