Romeo and Juliet Is EXPTIME-Complete

Harmender Gahlawat
G-SCOP, Grenoble-INP, France

Jan Matyáš Křišťan
Faculty of Information Technology, Czech Technical University in Prague, Czech Republic

Tomáš Valla
Faculty of Information Technology, Czech Technical University in Prague, Czech Republic

Abstract

Romeo and Juliet is a two player Rendezvous game played on graphs where one player controls two agents, *Romeo* ($\mathcal{R}$) and *Juliet* ($\mathcal{J}$) who aim to meet at a vertex against k adversaries, called *dividers*, controlled by the other player. The optimization in this game lies at deciding the minimum number of dividers sufficient to restrict $\mathcal{R}$ and $\mathcal{J}$ from meeting in a graph, called the *dynamic separation number*. We establish that Romeo and Juliet is EXPTIME-complete, settling a conjecture of Fomin, Golovach, and Thilikos [Inf. and Comp., 2023] positively. We also consider the game for directed graphs and establish that although the game is EXPTIME-complete for general directed graphs, it is PSPACE-complete and co-W[2]-hard for directed acyclic graphs.

2012 ACM Subject Classification Mathematics of computing → Graph algorithms

Keywords and phrases Rendezvous Games on graphs, EXPTIME-completeness, Dynamic Separators

Digital Object Identifier 10.4230/LIPIcs.MFCS.2024.54

Funding This work was supported by the Czech Science Foundation Grant no. 24-12046S. This work was supported by the Grant Agency of the Czech Technical University in Prague, grant No. SGS23/205/OHK3/3T/18.

1 Introduction

The study of *Rendezvous Games* was initiated by Alpern [2] where two agents, that are randomly placed in some known search region and move at unit speed, aim to meet each other in least expected time. Since then, several variants of rendezvous games have been considered on graphs [3, 10, 24]. Fomin, Golovach, and Thilikos [12] introduced the rendezvous game on graph with *adversaries* where a team of *dividers* aim to prevent the meeting of two passionate lovers, say *Romeo* and *Juliet*. We refer to this game as Romeo and Juliet.

Romeo and Juliet is played on finite, connected, and undirected graphs between two players: *facilitator* and *divider*. The facilitator has two agents, *Romeo*, denoted by $\mathcal{R}$, and *Juliet*, denoted by $\mathcal{J}$, that start the game at two designated vertices s and t of G, respectively. The divider has k agents, $D_1, \dots D_k$, and their starting position is selected by the divider from vertices in $V(G) \setminus \{s,t\}$. Several divider agents can occupy the same vertex. Afterwards, the divider player and the facilitator player make alternate moves, starting with the facilitator. In a move, a player, for each of its agents, either moves the agent to an adjacent vertex not occupied by any agent of the other player or keeps it on the same vertex. A situation where $\mathcal{R}$ and $\mathcal{J}$ are on the same vertex is a *meet*. The facilitator wins if $\mathcal{R}$ and $\mathcal{J}$ meet, and the divider wins if it succeeds in preventing the meet of $\mathcal{R}$ and $\mathcal{J}$ forever. Accordingly, we have the following decision version of the problem. We define the game formally in Section 2.

> Romeo and Juliet
> **Input**: A graph G with two specified vertices s and t, and an integer $k \in \mathbb{N}$.
> **Question**: Can the facilitator win on G starting from s and t against the divider with k agents?

49th International Symposium on Mathematical Foundations of Computer Science (MFCS 2024).
Editors: Rastislav Královič and Antonín Kučera; Article No. 54; pp. 54:1–54:16
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

We consider the computational complexity of this game and establish that Romeo and Juliet is EXPTIME-complete, resolving a conjecture of Fomin, Golovach, and Thilikos [12] positively. We further define the game for directed graphs and extend our EXPTIME-completeness result to directed graphs as well. We then establish that the game stays PSPACE-complete for directed acyclic graphs.

Rules of Romeo and Juliet are very similar to the rule of classical Cops and Robber game introduced by Nowakowski and Winkler [20], and Quilliot [21]. Cops and Robber fall in the broad range of *Graph Searching*, where a a set of agents, called *pursuers*, plan to catch one or multiple *evaders* in a graph under some movement rules. We refer to the annotated bibliography by Fomin and Thilikos [13] and recent monographs [6, 7] for further references on this topic.

Observe that if $s = t$ or s and t are adjacent vertices, then the facilitator wins trivially. For distinct non-adjacent vertices s and t, let s, t-*dynamic separation number* be the minimum k such that k dividers have a winning strategy against $\mathcal{R}$ and $\mathcal{J}$ starting at s and t, respectively. Since the dividers have a winning strategy by placing a divider on each vertex of a minimum size s, t-vertex cut (for distinct and non-adjacent vertices), the s, t-dynamic separation number is a well-defined graph invariant for s, t.

Fomin, Golovach, and Thilikos [12] proved that Romeo and Juliet is PSPACE-hard for general graphs. They conjectured that the game is, in fact, EXPTIME-complete. We resolve their conjecture positively by providing the following theorem.

▶ **Theorem 1.** *Romeo and Juliet is EXPTIME-complete for undirected graphs.*

Fomin, Golovach, and Thilikos [12] gave a backtracking based $n^{\mathcal{O}(k)}$ time algorithm for Romeo and Juliet, which is also a $2^{\mathcal{O}(n \log n)}$ time algorithm. Hence, to prove the EXPTIME-completeness, we only need to prove EXPTIME-hardness of Romeo and Juliet. To this end, we provide a non-trivial reduction from GuardUndir (a *guarding game* on undirected graphs), which is known to be EXPTIME-complete [22].

Several graph-searching games have a natural generalization to directed graphs and are well-studied [4, 5, 8, 9, 15, 16, 17]. Romeo and Juliet can also be considered on directed graphs where the agents can only move along the orientations of the arcs. We begin by establishing that the Romeo and Juliet game stays EXPTIME-complete on directed graphs. To this end, we provide a rather easy and straightforward reduction from Guard (a guarding game on directed graphs), which is known to be EXPTIME-complete [23], to Romeo and Juliet on directed graphs.

▶ **Theorem 2.** *Romeo and Juliet is EXPTIME-complete for oriented graphs.*

Next, we consider Romeo and Juliet on directed acyclic graphs (DAGs). Fomin, Golovach, and Thilikos [12] also considered a variant of this game, Romeo and Juliet in Time, where the question is whether $\mathcal{R}$ and $\mathcal{J}$ can meet in at most τ rounds. They established that this game is PSPACE-hard and co-W[2]-hard parameterized by k (for undirected graphs). We provide a general framework to establish computational complexity results for Romeo and Juliet on DAGs. In particular, we define a relaxation of the game –Relaxed Romeo and Juliet– where the dividers have an added relaxation that they can move to a vertex occupied by $\mathcal{R}$ or $\mathcal{J}$, but $\mathcal{R}$ or $\mathcal{J}$ cannot finish a move by occupying the same vertex as a divider. We present a reduction from Romeo and Juliet in Time on general graphs to Romeo and Juliet on directed acyclic graphs. This helps us to translate the hardness results proved for Romeo and Juliet in Time on general graphs to Romeo and Juliet on DAGs. In particular, this establishes that Romeo and Juliet remains PSPACE-hard and co-W[2]-hard parameterized by k even on DAGs. To prove PSPACE-completeness for DAGs, we also provide a polynomial-space algorithm for Romeo and Juliet on DAGs. We show the following results.

▶ **Theorem 3.** *Romeo and Juliet is PSPACE-complete when restricted to directed acyclic graphs.*

▶ **Theorem 4.** *Romeo and Juliet is co-W[2]-hard parameterized by k when restricted to directed acyclic graphs.*

Brief Survey. The study of rendezvous games with adversaries was initiated by Fomin, Golovach, and Thilikos [12] and they conducted an extensive study of the computational complexity of this problem and established the following results. Romeo and Juliet, as well as Romeo and Juliet in Time, are PSPACE-hard and co-W[2]-hard parameterized by k, and both of these problems admit a $n^{\mathcal{O}(k)}$ algorithm. This algorithm is optimal in the sense that, assuming ETH , none of these problems can be solved in $n^{o(k)}$ time. Moreover, Romeo and Juliet in Time is co-NP-complete even for $\tau = 2$, and admit a FPT algorithm parameterized by τ and the neighbourhood diversity of the graph, combined. Interestingly, for chordal graphs and P_5-free graphs, the s, t-dynamic separation number is same as the minimum size of a s, t-vertex cut, which establishes that Romeo and Juliet is polynomial time solvable for these classes.

Misra et al. [18] conducted further analysis of this game from a parameterized complexity perspective and established the following interesting results. Romeo and Juliet is co-para-NP-hard parameterized by the treewidth of the input graph. Further, Romeo and Juliet remains co-W[1]-hard when parameterized by the feedback vertex set number and the solution size (combined), and when parameterized by the pathwidth and the solution size (combined). On the positive side, they established that Romeo and Juliet is FPT when parameterized by the vertex cover number and solution size (combined) by the design of an exponential kernel, and complemented this result by proving that it is unlikely to obtain a polynomial kernel by these parameters. Finally, Romeo and Juliet can be solved in polynomial time for treewidth-2 graphs and grids.

An important part of our result is related to the so-called *guarding game*, introduced by Fomin et al. [11], is played on a graph G by two alternating players, the *cop-player* and the *robber-player*, each having their pawns (c cops and one robber, respectively). The vertex set $V(G)$ is partitioned into a *cop region* C and a *robber region* $R = V(G) \setminus C$, and the goal of the cops is to prevent the robber, who starts at some vertex of R, from entering a vertex of C. The computational complexity of the guarding game depends heavily on the chosen restrictions on the graph G. In particular, if Robber's region (R) is only a path, then the problem can be solved in polynomial time, and when robber moves in a tree (or even in a star), then the problem is NP-complete, and if Robber is moving in a DAG, the problem becomes PSPACE-complete [11]. Later Fomin, Golovach and Lokshtanov [14] studied the *reverse guarding game* with the same rules as in the guarding game, except that the cop-player plays first. They proved that the related decision problem is PSPACE-hard on undirected graphs. Nagamochi [19] has also shown that that the problem is NP-complete even if C induces a 3-star and that the problem is polynomially solvable if R induces a cycle. Also, Reddy, Krishna and Rangan [25] proved that if the robber-region is an arbitrary undirected graph, then the decision problem is PSPACE-hard. Šámal and Valla established that the guarding game is, in fact, ETIME-complete under log-space reductions for both directed [23] as well undirected graphs [22].

Organization. We begin with formal preliminaries and definitions in Section 2. In Section 3 we establish the EXPTIME-completeness of Romeo and Juliet for undirected graphs. In Section 4, we extend our EXPTIME-completeness result to directed graphs. We conclude in Section 5. To respect the space restrictions the section concerning Romeo and Juliet on DAGs has been omitted.

2 Preliminaries

For $\ell \in \mathbb{N}$, let $[\ell] = \{1, \dots, \ell\}$.

Graph Theory. For a graph G, we denote its vertex set by $V(G)$ and edge set by $E(G)$. We denote the size of $V(G)$ by n and size of $E(G)$ by m. In this paper, we consider finite, connected, and simple graphs. Let v be a vertex of a graph G. Then, by $N(v)$ we denote the *open neighbourhood* of v, that is, $N(v) = \{u \mid uv \in E(G)\}$. By $N[v]$ we denote the *closed neighbourhood* of v, that is, $N[v] = N(v) \cup \{v\}$. For $X \subseteq V(G)$, we define $N_X(v) = N(v) \cap X$ and $N_X[v] = N[v] \cap X$. The *length* of a path or cycle is the number of edges in it. A *u, v-path* is a path with endpoints u and v. For $u, v \in V(G)$, let $d(u, v)$ denote the length of a shortest u, v-path. A path is *isometric* if it is a shortest path between its endpoints.

Computational complexity. The complexity class PSPACE is the set of all decision problems that can be solved by a Turing machine using a polynomial amount of space. The class EXPTIME (sometimes denoted EXP) is the set of all decision problems that are solvable by a deterministic Turing machine in the $O(2^{p(n)})$ time where $p(n)$ is a polynomial of n.

Romeo and Juliet. Romeo and Juliet is played on a graph G, where the input prescribes the number of dividers k, and the starting positions s_0 and t_0 of $\mathcal{R}$ and $\mathcal{J}$, respectively. The game starts with $\mathcal{R}$ and $\mathcal{J}$ occupying the initial vertices s_0, and t_0, respectively. Then, the divider player places its agents $D_1, \dots, D_k$ on vertices $d_0^1, \dots, d_0^k$, respectively, such that $\{d_0^1, \dots, d_0^k\} \cap \{s_0, t_0\} = \emptyset$. We call this state $\mathcal{S}_0 = (s_0, t_0, d_0^1, \dots, d_0^k)$. For $i \geq 0$, let $\mathcal{D}_i$ denote the set $\{d_i^1, \dots, d_i^k\}$. Multiple dividers may occupy the same vertex, and $|\mathcal{D}_i|$ may be less than k. After this, the game proceed in *rounds*, where each round consists of a divider move, followed by a facilitator move. In round i, $i \geq 1$, first the facilitator moves $\mathcal{R}$ to a vertex $s_i \in N[s_{i-1}] \setminus \mathcal{D}_{i-1}$ and $\mathcal{J}$ to a vertex $t_i \in N[t_{i-1}] \setminus \mathcal{D}_{i-1}$. Then, the divider moves each divider D_p, $p \in [k]$, to a vertex $d_i^p \in N[d_{i-1}^p] \setminus \{s_i, t_i\}$. This gives us a game state $\mathcal{S}_i = (s_i, t_i, d_i^1, \dots, d_i^k)$. If the facilitator can ensure that for some $i \geq 0$, $s_i = t_i$, we say that the facilitator has a winning strategy. On the other hand, if the divider player can ensure that for each $i \geq 0$, $s_i \neq t_i$, then the divider player has a winning strategy. For directed graphs, the rules are exactly the same with the only difference that the agents can only move along the orientations of the arcs. Fomin et al. [12] gave an algorithm for Romeo and Juliet with running time $\mathcal{O}(2^{\mathcal{O}(n \log n)})$. It is easy to see that the algorithm works for directed graphs as well. Hence, we have the following proposition.

▶ **Proposition 5** ([12]). *Romeo and Juliet is in class EXPTIME on directed as well as undirected graphs.*

We have the following trivial observation that shall be useful to us.

▶ **Observation 6.** *Let $\mathcal{S}_i = (s_i, t_i, d_i^1, \dots, d_i^k)$ be a game state in some graph G, and let $y \in V(G)$. If there is a s_i, y-path (resp., t_i, y-path) P such that (i) P is an isometric path of length ℓ, (ii) and for each vertex $v \in V(P)$ and $u \in \mathcal{D}_i$, $d(s_i, v) \leq d(u, v)$ (resp., $d(t_i, v) \leq d(u, v)$), then the facilitator player can ensure that $\mathcal{S}_{i+\ell} = (s_{i+\ell} = y, t_{i+\ell}, d_{i+\ell}^1, \dots, d_{i+\ell}^k)$ (resp., $\mathcal{S}_{i+\ell} = (s_{i+\ell}, t_{i+\ell} = y, d_{i+\ell}^1, \dots, d_{i+\ell}^k)$).*

Guarding Game. The GuardUndir game is played on an undirected graph G, where $V(G)$ is partitioned into two regions: *Cop region* $C \subset V(G)$, and *Robber region* $R = V(G) \setminus C$. There is a prescribed vertex $r_0 \in R$, where the robber starts. The games begins with the robber occupying the vertex r_0. Then, k cops occupy vertices $c_0^1, \dots, c_0^k$ such that for each $j \in [k]$, $c_0^j \in C$. More than one cop may occupy the same vertex. This gives us game state $\mathcal{G}_0 = (r_0, c_0^1, \dots, c_0^k)$. Let $\mathcal{C}_i$ denote the set of vertices $\{c_0^i, \dots, c_k^i\}$. Then, the game proceeds in rounds. In round i, $i > 0$, first the robber moves to a vertex $r_i \in N[r_{i-1}] \setminus \mathcal{C}_{i-1}$. Then, the cop player moves the cop C_j, $j \in [k]$, to a vertex $c_i^j \in N_C[c_{i-1}^j] \setminus \{r_i\}$. If the robber can ensure that for some $i \geq 0$, $r_i \in C$, then the robber has a winning strategy. Otherwise, if the cops can ensure that for each $i \geq 0$, $r_i \notin C$, then the cop player has a winning strategy.

It is worth noting that the GuardUndir game where the starting position of the robber is not specified is also studied. But for our purposes, we consider the variant where the starting position of the robber, i.e., r_0 is fixed. Furthermore, we assume that $G[C]$ as well as $G[R]$ are connected subgraphs of G.

The Guard game is analog of GuardUndir on directed graphs. The rules of the game are exactly the same as the undirected case with the only change that the agents can only move along the orientations of the arcs. Šámal and Valla [22, 23] proved the following.

▶ **Proposition 7** ([23, 22]). *Guard and GuardUndir are EXPTIME-complete.*

3 Romeo and Juliet is EXPTIME-complete

In this section, we establish that Romeo and Juliet is EXPTIME-complete for undirected graphs. To prove this, we provide a non-trivial reduction from GuardUndir to Romeo and Juliet on undirected graphs. We begin by providing an overview of our reduction.

Overview. First, we make two copies of the cop region C as $C_{\mathcal{R}}$ and $C_{\mathcal{J}}$, and two copies of the robber region R as $R_{\mathcal{R}}$ and $R_{\mathcal{J}}$ such that $\mathcal{R}$ starts in $R_{\mathcal{R}}$ and $\mathcal{J}$ starts in $R_{\mathcal{J}}$. We will have $2k$ dividers. Moreover, we use some gadgets, that we call *secret gardens*, which ensure that after each round, if each of $C_{\mathcal{R}}$ and $C_{\mathcal{J}}$ does not host at least k dividers each, $\mathcal{R}$ and $\mathcal{J}$ will meet in one of the secret gardens after a finite number of rounds. Moreover, given that both $C_{\mathcal{R}}$ and $C_{\mathcal{J}}$ host at least k dividers each, if $\mathcal{R}$ and $\mathcal{J}$ are able to enter the vertices of $C_{\mathcal{R}}$ and $C_{\mathcal{J}}$, respectively, then they will be able to meet in the next round. Hence, for dividers to win, the game is, more or less, similar to restricting $\mathcal{R}$ and $\mathcal{J}$ to enter $C_{\mathcal{R}}$ and $C_{\mathcal{J}}$, respectively, where both $C_{\mathcal{R}}$ and $C_{\mathcal{J}}$ host exactly k dividers each. Below, we provide our construction in detail.

Construction. Let (H, k, r) be an instance of the GuardUndir, where r is the starting position of the robber and $V(H)$ consists of the cop region C and the robber region $R = V(H) \setminus C$. We assume that $H[C]$ as well as $H[R]$ is connected. We will construct an instance $(G, 2k)$ of Romeo and Juliet in the following manner. Since the construction has several components, we will define the components of our construction (reduction) individually. Our graph G will have following components.

1. $C_{\mathcal{R}}$, $C_{\mathcal{J}}$, $R_{\mathcal{R}}$, and $R_{\mathcal{J}}$. We begin by constructing two copies of the graph H as $G_{\mathcal{R}}$ and $G_{\mathcal{J}}$, corresponding to $\mathcal{R}$ and $\mathcal{J}$, respectively. See Figure 1 for an illustration. More specifically, $G_{\mathcal{R}}$ (resp., $G_{\mathcal{J}}$) contains a copy $C_{\mathcal{R}}$ (resp., $C_{\mathcal{J}}$) of C and $R_{\mathcal{R}}$ (resp., $R_{\mathcal{J}}$) of R. Formally, $V(G_{\mathcal{R}}) = \{u_{\mathcal{R}} \mid u \in V(H)\}$ and, $V(G_{\mathcal{J}}) = \{u_{\mathcal{J}} \mid u \in V(H)\}$. Moreover,

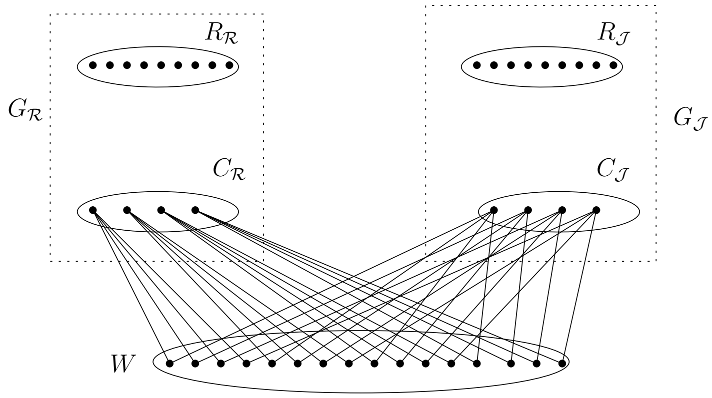

Figure 1 The graphs $G[R_{\mathcal{R}} \cup C_{\mathcal{R}}]$ (i.e., $G_{\mathcal{R}}$) as well as $G[R_{\mathcal{J}} \cup C_{\mathcal{J}}]$ (i.e., $G_{\mathcal{J}}$) are isomorphic to H. Here, we do not display edges in $G_{\mathcal{R}}$ and $G_{\mathcal{J}}$ to ease the illustration.

$E(G_{\mathcal{R}}) = \{u_{\mathcal{R}}v_{\mathcal{R}} \mid uv \in E(H)\}$ and $E(G_{\mathcal{J}}) = \{u_{\mathcal{J}}v_{\mathcal{J}} \mid uv \in E(H)\}$. Finally, the starting position of $\mathcal{R}$ is $r_{\mathcal{R}}$ and $\mathcal{J}$ is $r_{\mathcal{J}}$ (where r is the starting position of the robber in the GuardUndir), i.e., $s_0 = r_{\mathcal{R}}$ and $t_0 = r_{\mathcal{J}}$.

2. Connecting $C_{\mathcal{R}}$ to $C_{\mathcal{J}}$. For each vertex $x \in C_{\mathcal{R}}$ and each vertex $y \in C_{\mathcal{J}}$, we connect x and y using a path $P_{xy} = xw_{xy}y$ of length 2. Let W be the set of vertices that lie in the middle of these paths, i.e., $W = \{w_{xy} \mid x \in C_{\mathcal{R}}, y \in C_{\mathcal{J}}\}$. Moreover, each vertex of W has degree exactly 2 in G. See Figure 1 for an illustration. We have the following trivial observation, which follows from the fact that each vertex in W have degree exactly 2.

▶ **Observation 8.** *Consider a game state $S_i = \{s_i, t_i, d_i^1, \ldots, d_i^{2k}\}$, $i > 0$, such that $s_i \in C_{\mathcal{R}}$ and $t_i \in C_{\mathcal{J}}$, and $w_{s_it_i} \notin \mathcal{D}_{i-1}$, then $\mathcal{R}$ and $\mathcal{J}$ can meet at $w_{s_it_i}$ in the next round.*

Proof. The proof follows from the fact that if $w_{s_it_i} \notin \mathcal{D}_{i-1}$, then $w_{s_it_i} \notin \mathcal{D}_i$ since $w_{s_it_i}$ is connected only to s_i and t_i and $\{s_i, t_i\} \cap \mathcal{D}_i = \emptyset$. ◀

3. *Secret Gardens*. Next, we construct $2k + 2$ secret gardens $S_1, \ldots, S_{2k+2}$, the goal of which is to ensure that after each round, both $C_{\mathcal{R}}$ and $C_{\mathcal{J}}$ must host exactly k dividers each. See Figure 2 for an illustration. Each S_i, $i \in [2k + 2]$, consists of k independent vertices.

4. *Bridges*. Next, we construct $2k + 2$ *divider bridges* $B_1, \ldots B_{2k+2}$, where each B_i ($i \in [2k + 2]$) consists of k independent vertices, and $2k + 2$ *lover bridges* $L_1, \ldots, L_{2k+2}$, where each L_i consists of two independent vertices a_i and b_i. See Figure 2 for an illustration.

5. Connecting bridges and gardens. For each $i \in [2k + 2]$, $G[S_i \cup B_i]$ induces a complete bipartite graphs. Next, for $i \in [k + 1]$, we connect each vertex of S_i to a_i via an edge and each vertex of b_i to S_i via a path of length 2. Symmetrically, for $k + 2 \le i \le 2k + 2$, we connect each vertex of S_i to a_i via a path of length 2 and each vertex of S_i to b_i via an edge.

6. Connecting bridges to rest of the graph. Let β_H, β_C, and β_R be the diameter of graph H, graph $H[C]$, and graph $H[R]$, respectively. Then, let $\alpha = \max(\beta_H, \beta_C, \beta_R)$. For each vertex $x \in C_{\mathcal{R}}$ and each vertex $y \in \bigcup_{i\in[k+1]} B_i$, we connect x and y using a path of length α. Similarly, for each vertex $x \in C_{\mathcal{J}}$ and each vertex $y \in \bigcup_{k+1<i\le 2k+2} B_i$, we connect x and y using a path of length α. Next, for $i \in [k + 1]$, we connect each vertex $x \in R_{\mathcal{R}}$ to

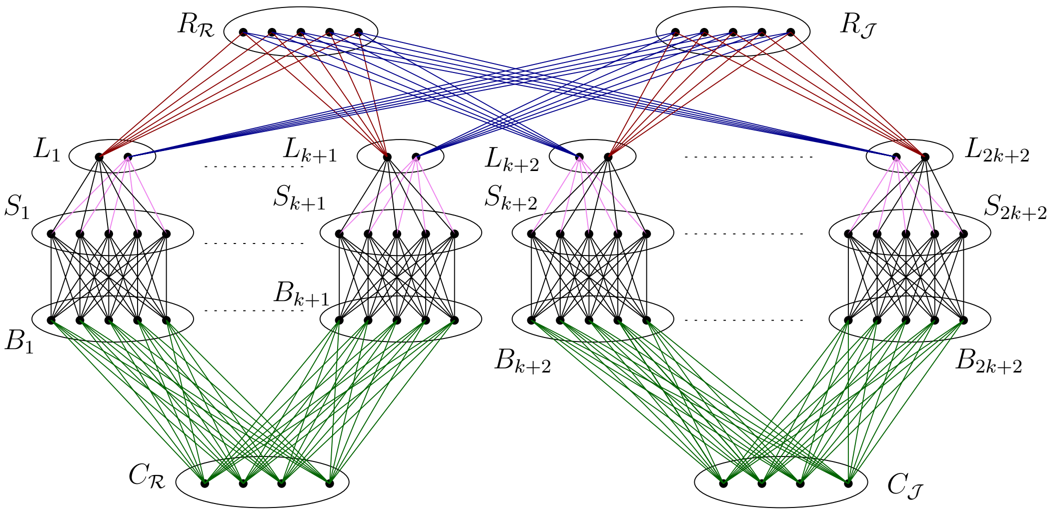

Figure 2 Illustration of secret gardens and bridges. Each black connection signifies an edge, blue, red, green, and violet connections signify paths of length $\alpha+2$, $\alpha+1$, α, and 2, respectively. Note that, due to our choice of α, no shortcuts are created in the original graph.

a_i using a path of length $\alpha+1$ and each vertex $y \in R_{\mathcal{J}}$ to b_i using a path of length $\alpha+2$. Symmetrically, for $k+2 \leq i \leq 2k+2$, we connect each vertex $x \in R_{\mathcal{R}}$ to a_i using a path of length $\alpha+2$ and each vertex $y \in R_{\mathcal{J}}$ to b_i using a path of length $\alpha+1$. See Figure 2 for an illustration. This completes our construction of G.

Subgraphs of G. For the ease of exposition, we name some of the induced subgraphs of G. Consider the induced subgraph $G' = G[V(G) \setminus (R_{\mathcal{R}} \cup R_{\mathcal{J}} \cup W)]$. Observe that G' has two connected components: one containing $C_{\mathcal{R}}$, say $G'_{\mathcal{R}}$, and the other containing $C_{\mathcal{J}}$, say $G'_{\mathcal{J}}$. Moreover, consider the induced subgraph $G'' = G'[V(G') \setminus (C_{\mathcal{R}} \cup C_{\mathcal{J}})]$. Observe that G'' contains exactly $2k+2$ connected components, say $G_1, \ldots, G_{2k+2}$, and let G_i be the connected component containing S_i (and hence, B_i and L_i).

The following observation follows directly from the construction of G

▶ **Observation 9.** *The following statements follow from the construction.*

1. *For distinct $i, j \in [2k+2]$, for a vertex $x \in V(G_i)$ and $y \in S_j$, $d(x,y) > \alpha+1$.*
2. *Let $x \in S_i$ for $i \in [k+1]$. For each vertex $y \in C_{\mathcal{R}}$, there is a x,y-path of length $\alpha+1$, and for each vertex $z \in C_{\mathcal{J}}$, there is a x,z-path of length $\alpha+3$.*
3. *Let $x \in S_i$ for $k+2 \leq i \leq 2k+2$. For each vertex $y \in C_{\mathcal{R}}$, there is a x,y-path of length $\alpha+3$, and for each vertex $z \in C_{\mathcal{J}}$, there is a x,z-path of length $\alpha+1$.*

The following lemma establishes that if $\mathcal{R}$ and $\mathcal{J}$ are in $R_{\mathcal{R}}$ and $R_{\mathcal{J}}$, respectively, then both $G'_{\mathcal{R}}$ and $G'_{\mathcal{J}}$ must host at least k dividers, else $\mathcal{R}$ and $\mathcal{J}$ meet in at most $\alpha+2$ rounds.

▶ **Lemma 10.** *Consider a game state $\mathcal{S}_i = (s_i, t_i, d_i^1, \ldots, d_i^{2k})$ for $i \geq 0$. If $s_i \in R_{\mathcal{R}}$ and $t_i \in R_{\mathcal{J}}$, and at least one of $G'_{\mathcal{R}}$ or $G'_{\mathcal{J}}$ hosts less than k dividers, then $s_{i+\alpha+3} = t_{i+\alpha+3}$.*

Proof. Without loss of generality, let us assume that $G'_{\mathcal{R}}$ hosts at most $k-1$ dividers at the end of round i (i.e., $\mathcal{D}_i \cap V(G'_{\mathcal{R}}) < k$). Therefore, at least one of $G_1, \ldots, G_{k+1}$ does not host any divider (as $G_1, \ldots, G_{k+1}$ are disjoint subgraphs of G'_R). Let G_p, $p \in [k+1]$, be such a component, i.e., $V(G_p) \cap \mathcal{D}_i = \emptyset$. To ease the presentation, let $x = s_i$ and $y = t_i$. Moreover, let P_x be the unique isometric x, a_p-path of length $\alpha+1$, and let P_y be the unique isometric

y, b_p-path of length $\alpha + 2$. Furthermore, let the vertices of S_p be marked as $v_1, \ldots, v_k$ and let, for $j \in [k]$, the (degree-2) vertex connecting v_j and b_p be u_j. We have the following crucial claim that proves our lemma.

▷ Claim 11. $\mathcal{R}$ and $\mathcal{J}$ can move along the paths P_x and P_y such that $s_{i+\alpha+2} \in S_p$, $t_{i+\alpha+2} = b_p$. Moreover, if $s_{i+\alpha+2} = v_j$, where $j \in [k]$, then $u_j \notin \mathcal{D}_{i+\alpha+2}$.

Proof of Claim. First, we establish that $\mathcal{R}$ can move to the vertex a_p in $\alpha + 1$ rounds. Due to Observation 6, to show that $\mathcal{R}$ can ensure that $s_{i+\alpha+1} = a_p$, it is sufficient to show that for each vertex $v \in V(P_x)$, $d(x, v) \leq \min_{u \in \mathcal{D}_i}(d(u, v))$. We distinguish the following cases depending on where u is in G.

1. $u \in C_{\mathcal{R}} \cup C_{\mathcal{J}} \cup W$: Observe that for each vertex $u \in C_{\mathcal{R}} \cup C_{\mathcal{J}} \cup W$ and each vertex $v \in V(P_x)$, $d(u, v) \geq \alpha + 1$ and $d(x, v) \leq \alpha + 1$. Hence, $d(x, v) \leq \min_{u \in \mathcal{D}_i}(d(u, v))$.
2. $u \in R_{\mathcal{R}} \cup R_{\mathcal{J}}$: Observe that for each vertex $u \in R_{\mathcal{R}} \cup R_{\mathcal{J}}$ and each vertex $v \in V(P_x)$, any u, v-path contains either x or a_p. In the former case, trivially $d(x, v) \leq d(u, v)$, and in the latter case, observe that $d(u, v) \geq \alpha + 1 \geq d(x, v)$. Hence, $d(x, v) \leq \min_{u \in \mathcal{D}_i}(d(u, v))$.
3. $u \in G_j$, for some $j \in [2k + 2]$: Due to our choice of p, clearly $u \notin V(G_p)$ (since we choose G_p such that $V(G_p) \cap \mathcal{D}_i = \emptyset$). Hence, $j \neq p$. Since G_j and G_p are disjoint components in $G[V(G) \setminus \{C_{\mathcal{R}} \cup C_{\mathcal{J}} \cup R_{\mathcal{R}} \cup R_{\mathcal{J}}\}]$, each u, v-path passes through a vertex of $u' \in C_{\mathcal{R}} \cup C_{\mathcal{J}} \cup R_{\mathcal{R}} \cup R_{\mathcal{J}}$, and hence this case is implied by the previous two cases.

Thus, the facilitator can ensure that $s_{i+\alpha+1} = a_p$. Finally, since $G'_{\mathcal{R}}$ hosts at most $k - 1$ dividers and for each vertex $u \in V(G) \setminus V(G'_{\mathcal{R}})$ and $v \in S_p$, $d(u, v) > \alpha + 1$, at most $k - 1$ dividers can reach the vertices of S_p in $\alpha + 1$ moves of dividers. Therefore, $|\mathcal{D}_{i+\alpha+1} \cap S_p| < k$. Hence, there is at least one $j \in [k]$ such that $v_j \in S_p \setminus \mathcal{D}_{i+\alpha+1}$, and hence $\mathcal{R}$ can move to v_j in this round, i.e., $s_{i+\alpha+2} = v_j \in S_p$.

The proof that of the claim $t_{i+\alpha+2} = b_p$ is symmetric to the proof that $s_{i+\alpha+1} = a_p$.

Next, we establish that $u_j \notin \mathcal{D}_{i+\alpha+2}$. Recall that $v_j \notin \mathcal{D}_{i+\alpha+1}$ and, due to our choice of p, $V(G_p) \cap \mathcal{D}_i = \emptyset$. It follows from our construction that for each vertex $w \in V(G) \setminus (G_p \cup C_{\mathcal{R}})$, $d(w, u_j) > \alpha+2$ and for $w' \in C_{\mathcal{R}}$, $d(w', u_j) = \alpha+2$ and each w', u_j-path of length $\alpha+2$ passes through v_j. Since $\mathcal{D}_i \cap V(G_p) = \emptyset$, if a divider, say D_ℓ, has to ensure that $d^\ell_{i+\alpha+2} = u_j$, D_ℓ has to ensure that $d^\ell_{i+\alpha+1} = v_j$, which is not possible since $v_j \notin \mathcal{D}_{i+\alpha+1}$. Hence, $u_j \notin \mathcal{D}_{i+\alpha+2}$. This completes the proof of our claim. ◁

Finally, since $s_{i+\alpha+2} = v_j \in S_p$ (for some $j \in [k]$), $t_{i+\alpha+2} = b_p$ and $u_j \notin \mathcal{D}_{i+\alpha+2}$ (Claim 11), the facilitator can ensure that $\mathcal{R}$ and $\mathcal{J}$ meet in the next round at u_j. ◀

Next, we have the following lemma which establishes that as long as $\mathcal{R}$ and $\mathcal{J}$ are in $R_{\mathcal{R}}$ and $R_{\mathcal{J}}$, respectively, both $C_{\mathcal{R}}$ and $C_{\mathcal{J}}$ must host exactly k dividers each.

▶ **Lemma 12.** *Consider a game state $\mathcal{S}_i = (s_i, t_i, d^1_i, \ldots, d^{2k}_i)$ for $i \geq 0$ such that $s_i \in R_{\mathcal{R}}$ and $t_i \in R_{\mathcal{J}}$. If $|\mathcal{D}_i \cap C_{\mathcal{R}}| \neq k$ or $|\mathcal{D}_i \cap C_{\mathcal{J}}| \neq k$, then the facilitator can ensure that $s_{i+\alpha+3} = t_{i+\alpha+3}$.*

Proof. The proof is similar to the proof of Lemma 10. Due to Lemma 10, we know that both $G'_{\mathcal{R}}$ and $G'_{\mathcal{J}}$ host exactly k dividers, and thus, $|\mathcal{D}_i \cap C_{\mathcal{R}}| \leq k$ and $|\mathcal{D}_i \cap C_{\mathcal{J}}| \leq k$, else $\mathcal{R}$ and $\mathcal{J}$ meet in $\alpha + 3$ rounds. At this point, let one of $C_{\mathcal{R}}$ or $C_{\mathcal{J}}$ hosts less that k dividers. Without loss of generality, let $|\mathcal{D}_i \cap C_{\mathcal{R}}| < k$. In this case, similarly to the proof of Lemma 10, there is at least one G_p, $p \in [k + 1]$, such that G_p does not contain any divider . Moreover, similarly to the proof of Lemma 10, only the dividers on $C_{\mathcal{R}}$ can reach S_p in $\alpha + 2$ rounds and hence at most $k - 1$ dividers can reach the vertices of S_p in $\alpha + 2$ rounds. Therefore, $\mathcal{R}$ and $\mathcal{J}$ have a strategy to ensure that $s_{i+\alpha+2} = z \in S_p$ and $t_{i+\alpha+2} = b_p$ such that the unique vertex on the isometric $s_{i+\alpha+2}, t_{i+\alpha+2}$-path does not any divider in this round. Hence, $s_{i+\alpha+3} = t_{i+\alpha+3}$. ◀

Next, we prove the following lemma which implies one side of our reduction.

▶ **Lemma 13.** *If the robber has a winning strategy in H against k cops, then $\mathcal{R}$ and $\mathcal{J}$ have a winning strategy in G against $2k$ dividers.*

Proof. The game starts with $\mathcal{R}$ and $\mathcal{J}$ placed on $r_{\mathcal{R}}$ and $r_{\mathcal{J}}$, respectively. Due to Lemma 12, we can assume that as long as $\mathcal{R}$ and $\mathcal{J}$ are in $R_{\mathcal{R}}$ and $R_{\mathcal{J}}$, respectively, both $C_{\mathcal{R}}$ and $C_{\mathcal{J}}$ must host exactly k dividers each, else $\mathcal{R}$ and $\mathcal{J}$ win. First, we will show that after a finite number of rounds $\mathcal{R}$ can enter $C_{\mathcal{R}}$. Observe that any move of dividers in $C_{\mathcal{R}}$ (in G) corresponds to a valid move of cops in C (in H) and $\mathcal{R}$ can make any move in $G[C_{\mathcal{R}} \cup R_{\mathcal{R}}]$ against dividers in G that is accessible to the robber in H against the corresponding position of the cops. Hence, using the winning strategy of the robber, $\mathcal{R}$ can enter $C_{\mathcal{R}}$ after a finite number of rounds (since $C_{\mathcal{R}}$ hosts exactly k dividers). Similarly, $\mathcal{J}$ can move to $C_{\mathcal{J}}$ in a finite number of rounds.

But, observe that since the dividers in $C_{\mathcal{R}}$ and $C_{\mathcal{J}}$ may be using different strategies, $\mathcal{R}$ and $\mathcal{J}$ may not be able to simultaneously move to $C_{\mathcal{R}}$ and $C_{\mathcal{J}}$, respectively. Hence, we distinguish the following two cases.

1. $\mathcal{R}$ and $\mathcal{J}$ move simultaneously to $x \in C_{\mathcal{R}}$ and $y \in C_{\mathcal{J}}$, respectively. Due to Lemma 12, we may assume that when $\mathcal{R}$ and $\mathcal{J}$ made this move, both $C_{\mathcal{R}}$ and $C_{\mathcal{J}}$ hosted exactly k dividers each, and hence no vertex of W was occupied by any divider. Hence, $\mathcal{R}$ and $\mathcal{J}$ can meet in the next round due to Observation 8.
2. $\mathcal{R}$ moves to $x \in C_{\mathcal{R}}$ while $\mathcal{J}$ is on some vertex $y \in R_{\mathcal{J}}$. Similarly to the previous case, due to Lemma 12, we may assume that when $\mathcal{R}$ and $\mathcal{J}$ made this move, both $C_{\mathcal{R}}$ and $C_{\mathcal{J}}$ hosted exactly k dividers each. In the next move of the dividers, observe that there will be at least one G_p, $p \in [k+1]$, such that G_p does not contain any dividers. In the next $\alpha + 2$ rounds, $\mathcal{J}$ moves towards the vertex b_p and $\mathcal{R}$ moves first towards a vertex in S_p, and then to a neighbour of b_p. We note that $\mathcal{R}$ and $\mathcal{J}$ can make these moves following the same arguments presented in the proof of Lemma 10. Hence, $\mathcal{R}$ and $\mathcal{J}$ meet in the next round since they are they are at adjacent vertices.
3. $\mathcal{J}$ moves to $x \in C_{\mathcal{J}}$ while $\mathcal{R}$ is on some vertex $y \in R_{\mathcal{R}}$. This case is symmetric to Case 2.

This completes our proof. ◀

To complete our reduction, we need to establish that if k cops have a winning strategy in H, then $2k$ dividers have a winning strategy in G. To aid the presentation of the proof of this lemma, we need the following notion of *safe states*.

Safe states. Consider the GuardUndir on graph H and consider some game state $\mathcal{G}_i = (r_i, c_i^1, \dots, c_i^k)$ such that $r_i \in R$. We say that $\mathcal{G}_i$ is a *safe state* if the cops have a strategy to ensure that robber cannot enter C in any round $i' > i$. Similarly, consider the game Romeo and Juliet on graph G and some game state $\mathcal{S}_j = (s_j, t_j, d_j^1, \dots, d_j^{2k})$. We say that $\mathcal{S}_j$ is *$\mathcal{R}$-safe state* (resp., *$\mathcal{J}$-safe state*) if (1) $s_j \in R_{\mathcal{R}}$ (resp., $t_j \in R_{\mathcal{J}}$), (2) $\{d_1^j, \dots, d_k^j\} \subseteq C_{\mathcal{R}}$ (resp., $\{d_j^{k+1}, \dots, d_j^{2k}\} \subseteq C_{\mathcal{J}}$), and (3) the dividers $D_1, \dots, D_k$ (resp., $D_{k+1}, \dots, D_{2k}$) can ensure that if $\mathcal{R}$ (resp., $\mathcal{J}$) is restricted to $G_{\mathcal{R}}$ (resp., $G_{\mathcal{J}}$), then $\mathcal{R}$ (resp., $\mathcal{J}$) cannot enter a vertex of $C_{\mathcal{R}}$ (resp., $C_{\mathcal{J}}$) in any round $j' > j$. We say that a game state $\mathcal{S}_j$ is *almost-$\mathcal{R}$-safe state* if $\mathcal{R}$ is in some G_i, $i \in [2k+2]$, and the dividers $D_1, \dots, D_k$ have a strategy that ensures that if for some $j' > j$, $s_{j'} \in V(G_{\mathcal{R}}) \cup V(G_{\mathcal{J}})$ (i.e., $\mathcal{R}$ enters a vertex in $V(G_{\mathcal{R}}) \cup V(G_{\mathcal{J}})$), then $\mathcal{S}_{j'}$ is a $\mathcal{R}$-safe state. We define *almost-$\mathcal{J}$-safe state* analogously. We have the following easy, but useful, observations.

▶ **Observation 14.** *Consider a game state $\mathcal{S}_j = (s_j, t_j, d_j^1, \ldots, d_j^{2k})$ of* Romeo and Juliet *on G such that $s_j \in R_{\mathcal{R}}$ and $\{d_j^1, \ldots, d_j^k\} \subseteq C_{\mathcal{R}}$. Now, consider a corresponding game state $\mathcal{G}_i = (r_i, c_i^1, \ldots, c_i^k)$ of* GuardUndir *on H such that if $s_j = u_{\mathcal{R}}$, then $r_i = u$, and if, for $\ell \in [k]$, $d_j^\ell = v_{\mathcal{R}}$, then $c_j^\ell = v$. If $\mathcal{G}_i$ is a safe state then $\mathcal{S}_j$ is a $\mathcal{R}$-safe state.*

Proof. To prove our statement, we need to show that if $\mathcal{R}$ is restricted to $G_{\mathcal{R}}$, then the dividers have a strategy to ensure that, for $j' > j$, $\mathcal{R}$ cannot enter a vertex of $C_{\mathcal{R}}$. The dividers $D_1, \ldots, D_k$ can do so by mimicking the winning strategy of the cops from H in the following manner. Whenever $\mathcal{R}$ moves from a vertex $u_{\mathcal{R}}$ to $w_{\mathcal{R}}$ in $G_{\mathcal{R}}$, we move the robber in H from u to w, and then each cop C_ℓ, for $\ell \in [k]$, moves as per its winning strategy. Notice that this gives us a safe state $\mathcal{S}_{i+1}$. Then, each divider D_ℓ, for $\ell \in [k]$, copies the move of C_i such that if C_i moved from v to x, then D_i moves from $v_{\mathcal{R}}$ to $x_{\mathcal{R}}$. Using this strategy, the dividers $D_1, \ldots, D_k$ can ensure that, as long as $\mathcal{R}$ is restricted to $G_{\mathcal{R}}$, $\mathcal{R}$ can never enter a vertex of $C_{\mathcal{R}}$. Hence, $\mathcal{S}_j$ is a $\mathcal{R}$-safe state. ◀

Analogously, we can have the following observation for $\mathcal{J}$-safe states.

▶ **Observation 15.** *Consider a game state $\mathcal{S}_j = (s_j, t_j, d_j^1, \ldots, d_j^{2k})$ of* Romeo and Juliet *on G such that $t_j \in R_{\mathcal{J}}$ and $\{d_j^{k+1}, \ldots, d_j^{2k}\} \subseteq C_{\mathcal{J}}$. Now, consider a corresponding game state $\mathcal{G}_i = (r_i, c_i^1, \ldots, c_i^k)$ of* GuardUndir *on H such that if $t_j = u_{\mathcal{J}}$, then $r_i = u$, and if, for $k+1 \leq \ell \leq 2k$, $d_j^\ell = v_{\mathcal{J}}$, then $c_j^{\ell-k} = v$. If $\mathcal{G}_i$ is a safe state then $\mathcal{S}_j$ is a $\mathcal{J}$-safe state.*

▶ **Observation 16.** *Let k cops have a winning strategy in H against the robber starting at r. Then, for each vertex $v \in R$ ($\subseteq V(H)$), there exists a set of (not necessarily distinct) vertices $u_1, \ldots, u_k \in C$ such that the game state $\mathcal{G} = (v, u_1, \ldots, u_k)$ is a safe state.*

Proof. Targeting contradiction, let there be a vertex $v \in C$ such that for every $u_1, \ldots, u_k \in C$ (possibly $u_i = u_j$ for distinct i, j), $\mathcal{G} = (v, u_1, \ldots, u_k)$ is a not a safe state. Let $d(r, v) = \ell \leq \alpha$. Then, the robber have a strategy to reach a game state $\mathcal{G}_\ell = (r_\ell = v, c_\ell^1, \ldots, c_\ell^k)$, which is not a safe state (by our contradiction assumption). Hence, the cops do not have any strategy that can restrict the robber to R for each round $\ell' > \ell$, a contradiction to the fact that k cops have a winning strategy in H against the robber starting at r. ◀

Next, we have the following lemma.

▶ **Lemma 17.** *Let k cops have a winning strategy in H against the robber starting at r. Moreover, let $\mathcal{S}_j = (s_j, t_j, d_j^1, \ldots, d_j^{2k})$ be a game state in G. Then, the following are true.*

1. *If $s_j = a_p$ for some $p \in [k+1]$ and each vertex of S_p is occupied by a divider from $D_1, \ldots, D_k$, then $\mathcal{S}_j$ is an almost-$\mathcal{R}$-safe state.*
2. *If $s_j = a_p$ for some $k+2 \leq p \leq 2k+2$, and each vertex of B_p is occupied by some divider from $D_1, \ldots, D_k$, then $\mathcal{S}_j$ is an almost-$\mathcal{R}$-safe state.*

Proof. First, we will show that if $s_j = a_p$ for some $p \in [k+1]$ and each vertex of S_p is occupied by a divider from $D_1, \ldots, D_k$, then $\mathcal{S}_j$ is an almost-$\mathcal{R}$-safe state. Let the vertices of S_p be marked $v_1, \ldots, v_k$, and without loss of generality, let us assume that $d_j^i = v_i$, for $i \in [k]$. The dividers will maintain the following invariant for $\ell \geq j$: $d(d_\ell^i, v_i) = d(s_\ell, a_p)$. (We have that $d(d_j^i, v_i) = d(s_j, a_p) = 0$.) This invariant will ensure that whenever $\mathcal{R}$ is at the vertex a_p, each vertex of S_p is occupied by some divider, and hence, $\mathcal{R}$ can never access a vertex of S_p. Let $j' > j$ be the smallest integer such that $s_{j'} \in V(G_{\mathcal{R}}) \cup V(G_{\mathcal{C}})$. If no such j' exists, then $\mathcal{S}_j$ is trivially almost-$\mathcal{R}$-safe state, and hence we assume that j' exists. To establish that $\mathcal{S}_j$ is an almost-$\mathcal{R}$-safe state, we need to show the following:

1. $s_{j'} \in R_{\mathcal{R}}$: This is easy to see since $S_p \cup R_{\mathcal{R}}$ separates a_p from each vertex in $R_{\mathcal{J}} \cup C_{\mathcal{R}} \cup C_{\mathcal{J}}$ and whenever $\mathcal{R}$ is at a_p, all vertices of S_p are occupied by the dividers.
2. $\mathcal{S}_{j'}$ is a $\mathcal{R}$-safe state: To ensure this, the dividers implement the following strategy which maintains the invariant $d(d^i_\ell, v_i) = d(s_\ell, a_p)$ (for $\ell \geq j$ and $i \in [k]$). Let $\gamma \geq j$ be the least integer such that $s_\gamma = a_p$ and $s_{\gamma+1} \neq a_p$. Observe that there is a unique vertex $v_{\mathcal{R}}$ in $R_{\mathcal{R}}$ such that $d(s_{\gamma+1}, v_{\mathcal{R}}) = \alpha$. (For every other vertex $w_{\mathcal{R}} \in R_{\mathcal{R}} \setminus \{v_{\mathcal{R}}\}$, $\alpha + 1 \leq d(s_{\gamma+1}, v_{\mathcal{R}}) \leq \alpha + 2$). Since k cops have a winning strategy in H, due to Observation 16, we know that there exists a safe state $\mathcal{G} = (v, u_1, \ldots, u_k)$ in H such that $u_1, \ldots, u_k \in C$. Moreover, due to Observation 14, we know that, a game state $\mathcal{S}_{j''} = (s_{j''} = v_{\mathcal{R}}, t_{j''}, d^1_{j''}, \ldots, d^{2k}_{j''})$ such that, for $i \in [k]$, $d^i_{j''} = u_{i\mathcal{R}}$ is a $\mathcal{R}$-safe state. Now, each divider D_i chooses chooses a $v_i, u_{i\mathcal{R}}$-path, say P_i, of length $\alpha + 1$ and move on this path to maintain the invariant $d(d^i_\ell, v_i) = d(s_\ell, a_p)$ (for $\ell \geq j$ and $i \in [k]$). Now, we distinguish the following two cases:
   a. For $\gamma < \gamma' < j'$, $s_{\gamma'} \neq a_p$. In this case, observe that $s_{j'} = v_{\mathcal{R}}$ and $d^i_{j'} = u_{i\mathcal{R}}$, for $i \in [k]$, which gives a $\mathcal{R}$-safe state.
   b. There is some $\gamma < \gamma' < j'$, such that $s_{\gamma'} = a_p$. Observe that the game state $\mathcal{S}_{\gamma'}$ is identical to the state $\mathcal{S}_j$ from the perspective of $\mathcal{R}$ and $D_1, \ldots, D_k$ (due to our invariant). Hence, the dividers $D_1, \ldots, D_k$ can restart their strategy that they had starting from $\mathcal{S}_j$.

Second, we will establish that if $s_j = a_p$ for some $k+2 \leq p \leq 2k+2$ and each vertex of B_p is occupied by a divider from $D_1, \ldots, D_k$, then $\mathcal{S}_j$ is an almost-$\mathcal{R}$-safe state. It follows directly from our construction of G that for each vertex $y \in C_{\mathcal{R}}$ and $z \in B_p$, there is a y, z-path of length $\alpha + 2$ (that passes through W and $C_{\mathcal{J}}$). Similarly, for each vertex $z \in R_{\mathcal{R}}$, there is a a_p, z-path of length $\alpha + 2$. The proof is similar to the proof of the first case and hence we will provide the proof in a succinct manner. Let the vertices of B_p be marked $v_1, \ldots, v_k$, and without loss of generality let us assume that $d^i_j = v_i$ (for $i \in [k]$). Furthermore, let the vertices of S_p be marked $x_1, \ldots, x_k$ and let the unique vertex connected a_p and x_i be y_i. Now, the dividers will use the following strategy for $j' \geq j$ while always maintaining the invariant: $d(s_{j'}, a_p) = d(d^i_{j'}, v_i)$ for $i \in [k]$.

- If $s_{j'} = y_q$ for some $q \in [k]$, then for each $i \in [k]$, the divider D_i moves to the vertex x_i, i.e., $d^i_{j'} = x_i$. Observe that this ensures that $\mathcal{R}$ can never reach a vertex of S_p, and hence, whenever $\mathcal{R}$ will enter a vertex of $C_{\mathcal{R}} \cup C_{\mathcal{J}} \cup R_{\mathcal{R}} \cup R_{\mathcal{J}}$, it will do so at a vertex of $R_{\mathcal{R}}$.
- If $s_{j'-1} = a_p$ and $s_{j'} \notin \{a_p\} \cup \{y_1, \ldots, y_p\}$. Then, let v be the unique vertex in $R_{\mathcal{R}}$ such that $d(v, s_{j'}) = \alpha + 1$. Since k cops have a winning strategy in H, due to Observation 16, we know that there exists a safe state $\mathcal{G} = (v, u_1, \ldots, u_k)$ in H such that $u_1, \ldots, u_k \in C$. Moreover, due to Observation 14, we know that, a game state $\mathcal{S}_{j''} = (s_{j''} = v_{\mathcal{R}}, t_{j''}, d^1_{j''}, \ldots, d^{2k}_{j''})$ such that, for $i \in [k]$, $d^i_{j''} = u_{i\mathcal{R}}$ is a $\mathcal{R}$-safe state. Now, each divider D_i chooses chooses a $v_i, u_{i\mathcal{R}}$-path, say P_i, of length $\alpha + 2$ and move on this path to maintain the invariant $d(d^i_\ell, v_i) = d(s_\ell, a_p)$ (for $\ell \geq j$ and $i \in [k]$). If $\mathcal{R}$ reaches the vertex $v_{\mathcal{R}}$, then observe that we have reached a safe state. Otherwise, if $\mathcal{R}$ reaches to a_p in some round $\gamma > j'$, then we restart our strategy.

This completes our proof. ◀

Next, we have the following lemma, whose proof is identical to the proof of Lemma 17.

▶ **Lemma 18.** *Let k cops have a winning strategy in H against the robber starting at r. Moreover, let $\mathcal{S}_j = (s_j, t_j, d^1_j, \ldots, d^{2k}_j)$ be a game state in G. Then, the following are true.*

1. *If $t_j = a_p$ for some $p \in [k+1]$ and each vertex of B_p is occupied by a divider from $D_{k+1}, \ldots, D_{2k}$, then $\mathcal{S}_j$ is an almost-$\mathcal{J}$-safe state.*
2. *If $s_j = a_p$ for some $k+2 \leq p \leq 2k+2$, and each vertex of S_p is occupied by some divider from $D_{k+1}, \ldots, D_{2k}$, then $\mathcal{S}_j$ is an almost-$\mathcal{J}$-safe state.*

The following observation is directly implied by our construction.

▶ **Observation 19.** *Consider the connected components of the graph $G[V(G) \setminus (C_{\mathcal{R}} \cup C_{\mathcal{J}} \cup S_1 \cup \cdots \cup S_{2k})]$. The components containing $R_{\mathcal{R}}$ and the component containing $R_{\mathcal{J}}$ are disjoint, and let them be named $F_{\mathcal{R}}$ and $F_{\mathcal{J}}$, respectively.*

Due to Observation 19, if we can show that k of the dividers, say $D_1, \ldots, D_k$, can restrict $\mathcal{R}$ from entering any vertex of $C_{\mathcal{R}} \cup C_{\mathcal{J}} \cup S_1 \cup \cdots \cup S_{2k}$, and the other k dividers, say $D_{k+1}, \ldots, D_{2k}$ can restrict $\mathcal{J}$ from entering any vertex of $C_{\mathcal{R}} \cup C_{\mathcal{J}} \cup S_1 \cup \cdots \cup S_{2k}$, then $\mathcal{R}$ and $\mathcal{J}$ never be able to meet as they will be restricted to disjoint subgraphs $F_{\mathcal{R}}$ and $F_{\mathcal{J}}$, respectively, of G. We use a similar strategy to prove the following lemma, which proves the other side of our reduction.

▶ **Lemma 20.** *If k cops have a winning strategy in H against the robber starting at r, then $2k$ dividers have a winning strategy in G against $\mathcal{R}$ and $\mathcal{J}$ starting at $r_{\mathcal{R}}$ and $r_{\mathcal{J}}$, respectively.*

Proof. Since k cops have a winning strategy against the robber starting at r in H, there is a safe state $\mathcal{G}_0 = (r_0 = r, c_0^1, \ldots, c_0^k)$. Now, we begin ROMEO AND JULIET on G with the game state $\mathcal{S}_0 = (s_0 = r_{\mathcal{R}}, t_0 = r_{\mathcal{J}}, d_0^1, \ldots, d_0^{2k})$ such that if $c_0^\ell = v$, for $\ell \in [k]$, then $d_0^\ell = v_{\mathcal{R}}$ and $d_0^{\ell+k} = v_{\mathcal{J}}$. It follows from Observations 14 and 15 that $\mathcal{S}_0$ is $\mathcal{R}$-safe state as well as $\mathcal{J}$-safe state. Therefore, as long as $\mathcal{R}$ (resp., $\mathcal{J}$) is restricted to $G_{\mathcal{R}}$ (resp., $G_{\mathcal{J}}$), they cannot enter a vertex of $C_{\mathcal{R}}$ (resp., $C_{\mathcal{J}}$). Hence, unless at least one of $\mathcal{R}$ or $\mathcal{J}$ leaves $G_{\mathcal{R}}$ or $G_{\mathcal{J}}$, respectively, to move to some G_i, for $i \in [2k+2]$, they cannot meet. First, we prove the following claim.

▷ Claim 21. Let $\mathcal{S}_j$ be a $\mathcal{R}$-safe state such that $s_{j+1} \notin R_{\mathcal{R}}$. Then, for $j' > j$, the dividers $D_1, \ldots, D_k$ have a strategy that can ensure the following:

1. $s_{j'} \in F_{\mathcal{R}}$.
2. If $s_{j'} \in R_{\mathcal{R}}$, then $\mathcal{S}_{j'}$ is a $\mathcal{R}$-safe state.

Proof of Claim. Let P_x be the unique x, a_p-path that contains s_{j+1}. For $\ell \in [k]$, let $d_j^\ell = y_\ell$ (i.e., in round j, the divider D_ℓ occupies the vertex $y_\ell \in C_{\mathcal{R}}$). We distinguish the following two cases depending on whether $p \leq k$ or $p > k$.

Case 1: $p \leq k$. Let the vertices of S_p be marked $v_1, \ldots, v_k$. It follows from the construction that $d(y_\ell, v_\ell) = \alpha + 1$ and $d(x, a_p) = \alpha + 1$. Each D_ℓ chooses a y_ℓ, v_ℓ-path of length $\alpha + 1$, say P_ℓ, and move along it to maintain the following invariant for $j' > j$: $d(s_{j'}, a_p) = d(d_{j'}^\ell, v_\ell)$. Again, we have the following cases depending on the moves of $\mathcal{R}$.

1. $\mathcal{R}$ never reaches the vertex a_p: In this case observe that when $\mathcal{R}$ moves to $R_{\mathcal{R}}$, say in round j', it can only move to the vertex x (i.e., $s_{j'} = x$). In this case, observe that $d_{j'}^\ell = y_\ell$. Since this state is identical to $\mathcal{S}_j$ with respect to the placement of $\mathcal{R}$ and the dividers $D_1, \ldots D_k$, $\mathcal{S}_{j'}$ is a $\mathcal{R}$-safe state. Moreover, it is easy to see that $\mathcal{R}$ was restricted to $F_{\mathcal{R}}$ for each round $j \leq j'' \leq j'$.
2. $\mathcal{R}$ reaches the vertex a_p in some round j'': In this case, observe that each vertex v_ℓ of S_p is occupied by the divider D_ℓ (due to our invariant), and hence, $\mathcal{R}$ cannot move to a vertex of S_p in round $j'' + 1$. Observe that the game state $\mathcal{S}_{j''}$ is an almost-$\mathcal{R}$-safe state due to Lemma 17. Therefore $\mathcal{R}$ is restricted to $F_{\mathcal{R}}$ and whenever $\mathcal{R}$ reaches a vertex of $R_{\mathcal{R}}$ in some round j', then $\mathcal{S}_{j'}$ is a $\mathcal{R}$-safe state due to the definition of almost-$\mathcal{R}$-safe state.

Case 2: $p > k$. The proof of this case is similar to the proof of Case 1. Let the vertices of B_p be marked $v_1, \dots, v_k$. It follows from the construction that $d(y_\ell, v_\ell) = \alpha + 2$ and $d(x, a_p) = \alpha + 2$. Each D_ℓ chooses a y_ℓ, v_ℓ-path of length $\alpha + 2$, say P_ℓ, and move along it to maintain the following invariant for $j' > j$: $d(s_{j'}, a_p) = d(d^\ell_{j'}, v_\ell)$. If $\mathcal{R}$ never reaches the vertex a_p, then our invariant implies that whenever $\mathcal{R}$ will enter $R_\mathcal{R}$, it will enter at a safe state. If $\mathcal{R}$ reaches the vertex a_p, then each vertex of B_p is occupied by a divider (due to our invariant), which is an almost-$\mathcal{R}$-safe state due to Lemma 17. Hence, if $\mathcal{R}$ enters a vertex $R_\mathcal{R}$ in some round j', then $\mathcal{S}_{j'}$ is a $\mathcal{R}$-safe state. Further, observe that in all these rounds, $\mathcal{R}$ is restricted to $F_\mathcal{R}$. This completes the proof of this case. ◁

Next, we have the following claim whose proof is symmetric to the proof of Claim 21.

▷ Claim 22. Let $\mathcal{S}_j$ be a $\mathcal{J}$-safe state such that $t_{j+1} \notin R_\mathcal{J}$. Then, for $j' > j$, the dividers $D_{k+1}, \dots, D_{2k}$ have a strategy that can ensure the following:

1. $t_{j'} \in F_\mathcal{J}$.
2. If $t_{j'} \in R_\mathcal{J}$, then $\mathcal{S}_{j'}$ is a $\mathcal{J}$-safe state.

Finally, our proof follows from the following facts. Since we start from a state that is $\mathcal{R}$-safe state as well as $\mathcal{J}$-safe state, as long as $\mathcal{R}$ (resp., $\mathcal{J}$) is restricted to $G_\mathcal{R}$ (resp., $G_\mathcal{J}$), they cannot enter a vertex of $C_\mathcal{R}$ (resp., $C_\mathcal{J}$). Moreover, even if $\mathcal{R}$ (resp., $\mathcal{J}$) leave $G_\mathcal{R}$ (resp., $G_\mathcal{J}$), it is restricted to $F_\mathcal{R}$ (resp. $F_\mathcal{J}$), and whenever $\mathcal{R}$ (resp., $\mathcal{J}$) return to $G_\mathcal{R}$ (resp., $G_\mathcal{J}$), it returns to a $\mathcal{R}$-safe state (resp., $\mathcal{J}$-safe state), due to Claim 21 (resp., Claim 22). Since this strategy restricts $\mathcal{R}$ and $\mathcal{J}$ to two disjoint subgraphs $F_\mathcal{R}$ and $F_\mathcal{J}$, respectively, of G, $\mathcal{R}$ and $\mathcal{J}$ will never be able to meet. Hence, $2k$ dividers have a winning strategy in G against $\mathcal{R}$ and $\mathcal{J}$ starting at $r_\mathcal{R}$ and $r_\mathcal{J}$, respectively. This completes our proof. ◀

Finally, we haven the following theorem due to our construction of G from H, Propositions 5 and 7, and Lemmas 13 and 20.

▶ **Theorem 1.** *Romeo and Juliet is EXPTIME-complete for undirected graphs.*

4 EXPTIME-Completeness for Directed Graphs

In this section, we establish that Romeo and Juliet is EXPTIME-complete on directed graphs. Due to Proposition 5, to complete our proofs, we only need to establish that Romeo and Juliet is EXPTIME-hard. To this end, we will reduce Guard to Romeo and Juliet on directed graphs. This is a rather straightforward and easy construction.

Construction. Let $(\overrightarrow{G}, k, r)$ be an instance of Guard (where r is the starting position of the robber and $V(\overrightarrow{G})$ consists of the $C \cup R$). We construct an instance $(\overrightarrow{H}, k, s, t)$ in the following manner. Let $V(\overrightarrow{H}) = V(\overrightarrow{G}) \cup \{s, t, d_1, \dots, d_k\}$, and let $D = \{d_1, \dots, d_k\}$. Next, $E(\overrightarrow{H}) = E(\overrightarrow{G}) \cup \{\overrightarrow{sr}, \overrightarrow{sd_1}, \dots, \overrightarrow{sd_k}, \overrightarrow{d_1t}, \dots, \overrightarrow{d_kt}\}$. Moreover, for each d_i, $i \in [k]$, and $v \in C$, we add an arc $\overrightarrow{d_ic}$. Furthermore, for each vertex $u \in C$, we add an arc $\overrightarrow{ut}$. See Figure 3 for an illustration. Finally, the starting position for $\mathcal{R}$ is s and for $\mathcal{J}$ is t. This completes the construction. The following observation follows directly from our construction.

▶ **Observation 23.** *The following statements are true.*

1. *The vertex t is a sink, and hence $\mathcal{J}$ cannot move throughout the game.*
2. *If at any point in the game $\mathcal{R}$ is on a vertex in C, then $\mathcal{R}$ and $\mathcal{J}$ meet in the next step.*
3. *If there are less than k dividers, then $\mathcal{R}$ and $\mathcal{J}$ meet in at most 2 rounds.*

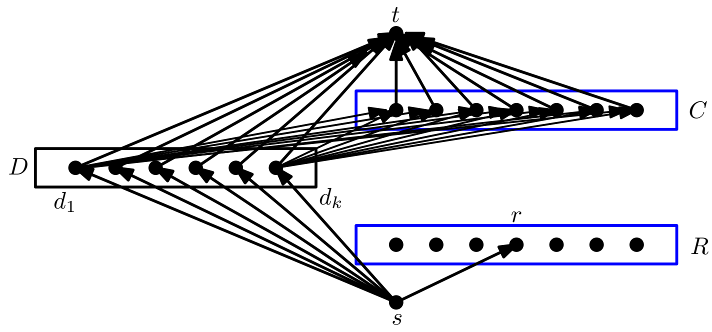

Figure 3 Here, we do not show edges of $\overrightarrow{G}$ to ease the presentation.

Proof. The proof of (1) and (2) follows directly from the construction and the game definition. To see the proof of (3), observe that if there are less than k dividers, there is at least one vertex in D, say x, that is not occupied by any of the dividers (since $|D| = k$). Hence, $\mathcal{R}$ can move to x in the first round, and since $\overrightarrow{xt}$ is an edge, $\mathcal{R}$ and $\mathcal{J}$ meet in the next round. ◀

The following lemma proves the soundness of our reduction.

▶ **Lemma 24.** *In $\overrightarrow{G}$, k cops have winning strategy against the robber starting at r if and only if k dividers have a winning strategy in $\overrightarrow{H}$ against $\mathcal{R}$ and $\mathcal{J}$ starting at s and t, respectively.*

Proof. In one direction, let k cops have a strategy to prevent the robber, who starts at r, from entering C. Then, we prove that k dividers have a winning strategy $\overrightarrow{H}$ against $\mathcal{R}$ and $\mathcal{J}$ starting at s and t, respectively. The k dividers begin with occupying each vertex of D. This restricts $\mathcal{R}$ from entering a vertex of D. Recall that $\mathcal{J}$ cannot move throughout the game (due to Observation 23). Now, the only move possible for $\mathcal{R}$ is to move to vertex r from s. Whenever $\mathcal{R}$ moves to r, the dividers move to the vertices in C where the cops begin in their winning strategy in $\overrightarrow{G}$. Now, observe that $\mathcal{R}$ cannot access s or any vertex in D. Hence, the dividers can restrict $\mathcal{R}$ and $\mathcal{J}$ from meeting by simply restricting $\mathcal{R}$ to ever enter C. Note that the the dividers can do so following the winning strategy for the cops as the rules of movement are the same for both of the games. Hence, k dividers have a winning strategy in $\overrightarrow{H}$ against $\mathcal{R}$ and $\mathcal{J}$ starting at s and t, respectively.

In the other direction, let the robber has a strategy to enter C in G starting from r. In this case, observe that in the beginning, k dividers have to occupy the vertices of D, else $\mathcal{R}$ and $\mathcal{J}$ meet in at most two steps ($\mathcal{R}$ moves to a vertex in D and then meet $\mathcal{J}$ in the next round). Now, $\mathcal{R}$ moves to r from s. Now, at most k dividers move to C. Since the dividers can make the same moves as cops and the robber have a winning strategy against any strategy of k cops to enter C starting from r, $\mathcal{R}$ can use the same strategy to enter C. Hence, $\mathcal{R}$ can enter C after a finite number of rounds, and then meet $\mathcal{J}$ in the next round at t (due to Observation 23). ◀

Hence, we have the following theorem as a consequence of our reduction, Propositions 5 and 7, and Lemma 24.

▶ **Theorem 2.** *Romeo and Juliet is EXPTIME-complete for oriented graphs.*

5 Conclusion

In this work, we considered Romeo and Juliet on directed as well as undirected graphs and established that the game is EXPTIME-complete in both cases, and that the game remains PSPACE-complete even for directed acyclic graphs. Moreover, we defined a game Relaxed Romeo and Juliet that provides a framework for extending the hardness results of Romeo and Juliet in Time on undirected graphs to Romeo and Juliet on DAGs.

It may be an interesting question to figure out if Romeo and Juliet in Time is also EXPTIME-complete as conjectured by Fomin, Golovach, and Thilikos [12]. Moreover, it is known that, assuming ETH, Romeo and Juliet cannot be solved in $n^{o(k)}$ (i.e., $2^{o(k \log n)}$) time [12]. It might be interesting to see if this result can be extended to a lower bound of the form $2^{o(n)}$. Note that this result will be incomparable to the current known bound as k can be $\mathcal{O}(n)$. Aigner and Fromme [1] established that the cop number for planar graphs is at most 3 by the use of a guarding lemma that states that 1 cop can guard the vertices of an isometric path. It is easy to see that the dynamic separation number of planar graphs is unbounded, for eg., consider $K_{2,n}$ and let s, t be the vertices of the partition with two vertices. It might be interesting to figure out the computational complexity of Romeo and Juliet on planar graphs, and more generally, graphs on surfaces.

References

1. M. Aigner and M. Fromme. A game of cops and robbers. *Discrete Applied Mathematics*, 8(1):1–12, 1984.
2. S. Alpern. The rendezvous search problem. *SIAM Journal on Control and Optimization*, 33(3):673–683, 1995.
3. S. Alpern and S. Gal. *The theory of search games and rendezvous*, volume 55. Springer Science & Business Media, 2006.
4. J. Barát. Directed path-width and monotonicity in digraph searching. *Graphs and Combinatorics*, 22(2):161–172, 2006.
5. D. Berwanger, A. Dawar, P. Hunter, S. Kreutzer, and J. Obdržálek. The dag-width of directed graphs. *Journal of Combinatorial Theory Series B*, 104(4):900–923, 2012.
6. A. Bonato. *An invitation to pursuit-evasion games and graph theory*, volume 97. American Mathematical Society, 2022.
7. A. Bonato and P. Pralat. *Graph searching games and probabilistic methods*. Chapman and Hall/CRC, 2017.
8. P. Bradshaw, S. A. Hosseini, and J. Turcotte. Cops and robbers on directed and undirected abelian Cayley graphs. *European Journal of Combinatorics*, 97, 2021. `doi:10.1016/j.ejc.2021.103383`.
9. S. Das, H. Gahlawat, U. k. Sahoo, and S. Sen. Cops and robber on some families of oriented graphs. *Theoretical Computer Science*, 888:31–40, 2021.
10. A. Dessmark, P. Fraigniaud, D. R. Kowalski, and A. Pelc. Deterministic rendezvous in graphs. *Algorithmica*, 46:69–96, 2006.
11. F. Fomin, P. Golovach, A. Hall, M. Mihalák, E. Vicari, and P. Widmayer. How to guard a graph? *Algorithmica*, 61(4):839–856, 2011.
12. F. V. Fomin, P. A. Golovach, and D. M. Thilikos. Can romeo and juliet meet? or rendezvous games with adversaries on graphs. *Information and Computation*, 293:105049, 2023.
13. F. V. Fomin and D. M. Thilikos. An annotated bibliography on guaranteed graph searching. *Theoretical Computer Science*, 399(3):236–245, 2008.
14. F.V. Fomin, P.A. Golovach, and D. Lokshtanov. Guard games on graphs: Keep the intruder out! *Theoretical Computer Science*, 412(46):6484–6497, 2011.

15 T. Johnson, N. Robertson, P. D. Seymour, and R. Thomas. Directed tree-width. *Journal of Combinatorial Theory Series B*, 82(1):138–154, 2001.

16 W. B. Kinnersley. Cops and robbers is exptime-complete. *Journal of Combinatorial Theory Series B*, 111:201–220, 2015.

17 P. Loh and S. Oh. Cops and robbers on planar directed graphs. *Journal of Graph Theory*, 86(3):329–340, 2017.

18 N. Misra, M. Mulpuri, P. Tale, and G. Viramgami. Romeo and juliet meeting in forest like regions. In *42nd IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, 2022.

19 H. Nagamochi. Cop-robber guarding game with cycle robber-region. *Theoretical Computer Science*, 412:383–390, 2011.

20 R. Nowakowski and P. Winkler. Vertex-to-vertex pursuit in a graph. *Discrete Mathematics*, 43(2-3):235–239, 1983.

21 A. Quilliot. *Thése d'Etat*. PhD thesis, Université de Paris VI, 1983.

22 R. Šámal, R. Stolař, and T. Valla. Complexity of the cop and robber guarding game. In *Combinatorial Algorithms: 22nd International Workshop, IWOCA 2011, Victoria, BC, Canada, July 20-22, 2011, Revised Selected Papers 22*, pages 361–373. Springer, 2011.

23 R. Šámal and T. Valla. The guarding game is E-complete. *Theoretical Computer Science*, 521:92–106, 2014.

24 A. Ta-Shma and U. Zwick. Deterministic rendezvous, treasure hunts, and strongly universal exploration sequences. *ACM Transactions on Algorithms (TALG)*, 10(3):1–15, 2014.

25 T. V. Thirumala Reddy, D. Sai Krishna, and C. Pandu Rangan. The guarding problem – complexity and approximation. In *Lecture Notes in Computer Science*, volume 5874, pages 460–470. Springer, 2009.