Guard Games on Graphs: Keep the Intruder Out!

Fedor V. Fomin* Petr A. Golovach* Daniel Lokshtanov*

Abstract

A team of mobile agents, called guards, tries to keep an intruder out of an assigned area by blocking all possible attacks. In a graph model for this setting, the agents and the intruder are located on the vertices of a graph, and they move from node to node via connecting edges. The area protected by the guards is a subgraph of the given graph. We investigate the algorithmic aspects of finding the minimum number of guards sufficient to patrol the area. We show that this problem is PSPACE-hard in general and proceed to investigate a variant of the game where the intruder must reach the guarded area in a single step in order to win. We show that this case approximates the general problem, and that for graphs without cycles of length 5 the minimum number of required guards in both games coincides. We give a polynomial time algorithm for solving the one-step guarding problem in graphs of bounded treewidth, and complement this result by showing that the problem is $W[1]$-hard parameterized by the treewidth of the input graph. We conclude the study of the one-step guarding problem in bounded treewidth graphs by showing that the problem is fixed parameter tractable (FPT) parameterized by the treewidth and maximum degree of the input graph. Finally, we turn our attention to a large class of sparse graphs, including planar graphs and graphs of bounded genus, namely graphs excluding some fixed apex graph as a minor. We prove that the problem is FPT and give a PTAS on apex-minor-free graphs.

1 Introduction

An intruder is trying to enter an area patrolled by a team of mobile units, for example robots. The goal of the robots is to protect the assigned area by blocking all possible attacks. We model this problem as a variant of the classical Cops and Robbers game [1] and we borrow the Cops and Robbers terminology, calling the guarding agents *cops* and the intruder a *robber*. The game of Cops and Robbers is a pursuit-evasion game played on a graph, see [4, 21] for references on different pursuit-evasion and search games on graphs.

The study of *cop-robber guard games* was initiated by Fomin et al. [19]. The guard game is played on a graph G by two players, the *cop-player* and the *robber-player*. The graph G can be directed or undirected, but we only consider undirected graphs in this paper. Each player has *pawns*, the cop-player has *cops* and the robber-player has a *robber*, placed on the vertices of G. The aim of the cop-player is to prevent the robber from entering the *protected region* $C \subsetneq V$, also called the *cop-region*, and correspondingly the aim of the robber is to penetrate the protected region. The robber can not enter a vertex if it is occupied by a cop, and the cops guard the protected region C by blocking all vertices which the robber can use as entry points to C. We say that a cop *guards* the vertex v which he occupies.

The game is played in alternating turns. In their first move players choose their initial positions. The cops choose vertices inside C to occupy, and the robber chooses some vertex outside C to start in. In each subsequent turn the respective player can *move* each of his pawns to a vertex adjacent to the vertex the pawn occupies or leave the pawn in its current position. The cops are only allowed to move within the protected region C, and the robber can only move onto a vertex with no cops on it. At any time of the game both players know the positions of the cops and the robber in G. The guard game is a *robber-win* game if the robber-player can at some turn move the robber onto a vertex within C with no cop on it. In this case we say that the robber-player *wins* the game. Otherwise the cop-player can forever prevent the robber-player

*Department of Informatics, University of Bergen, N-5020 Bergen, Norway, emails: {fedor.fomin|peter.golovach|daniello}@ii.uib.no

from winning. In this case we say that the game is a *cop-win* game, that the cop-player *wins* the game and that the cop-player can *guard* C.

The only difference between the game considered in this article and the game studied in [19] is the order of turns. In [19] the robber had to make the first move while in the problem studied here the cop-player starts the game. Despite the similar settings, the difference between the two games can be tremendous even for very simple examples. For instance consider the graph G in Figure 1 consisting of two paths P_R and P_C connected by a perfect matching. The path P_C is the cop-region, and the task of the robber to enter P_C from P_R. If the robber starts first, then one cop is sufficient to guard C since the cop only needs to occupy the vertex in P_C which is matched to the vertex occupied by the robber after the robber-players move. If cops start first, their initial positions should form a dominating set of P_C because otherwise the robber player can start in a vertex adjacent to an undominated vertex in C and enter C on his next turn. Thus, to protect P_C in the "cops-first" variant of the game we need at least $\lceil (V(P_C) - 2)/3 \rceil$ cops. The algorithmic behavior of the two problems is also quite different. It was proved in [19] that when the robber's territory is a path, the "robber-first" variant of the game is solvable in polynomial time. In contrast, a simple reduction from the minimum dominating set problem shows that "cops-first" variant is NP-hard, see Proposition 2.

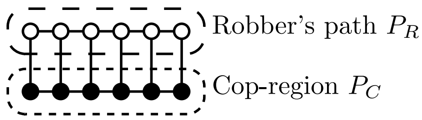

Figure 1: Paths P_C and P_R connected by a matching

A different well-studied problem, the ETERNAL DOMINATION problem which also is known as ETERNAL SECURITY is strongly related to the guard game. In the ETERNAL DOMINATION the objective is to place the minimum number of guards on the vertices of a graph G such that the guards can protect the vertices of G from an infinite sequence of attacks. In response to an attack of an unguarded vertex v, at least one guard must move to v and the other guards can either stay put, or move to adjacent vertices. Different variants of this problem have been considered in [5, 13, 12, 23, 25, 29, 28, 27]. The ETERNAL DOMINATION problem is a special case of our game. This can be seen as follows. Let G be a graph on n vertices, we construct a graph H from G by adding a clique K on n vertices and connecting the clique and G by n edges which form a perfect matching. If the cop-region of H is $V(G)$ then G has an eternal dominating set of size k if and only if k cops can guard $V(G)$.

Our results. In this paper we prove a number of algorithmic and complexity results about the guarding problem. We start with a proof that the problem is PSPACE-hard on *undirected* graphs. While many games are known to be PSPACE-hard, all known PSPACE-hardness results for cops and robbers, or pursuit evasion games are for the directed graph variant of the games [19, 24]. For example, the classical Cop and Robbers game was shown to be PSPACE-hard on directed graphs by Goldstein and Reingold in 1995 [24] whereas for undirected graphs, even an NP-hardness result was not known until very recently [20].

We show that the number of cops required to guard a graph is at most twice the number of cops required to protect the graph in the one-step variant of the game, that is when all players only make one move after the initial placement step. We show that this game is not only a good approximation of the general problem, but that for many graph classes like graphs without cycles of length 5 the two games are equivalent. We provide a number of FPT algorithms and parameterized complexity results for the one-step guarding problem. Our results include a polynomial time algorithm for the problem in graphs of bounded treewidth, a complexity result showing that our algorithm is essentially optimal, and an FPT algorithm for the problem parameterized by the treewidth and maximum degree of the input graph. Finally we use our treewidth-based algorithms to show that on graphs excluding some fixed apex graph as a minor the one-step guarding problem is FPT and admits a PTAS.

2 Definitions and preliminaries

We consider finite undirected graphs without loops or multiple edges. The vertex set of a graph G is denoted by $V(G)$ and its edge set by $E(G)$, or simply by V and E if this does not create confusion. If $U \subseteq V(G)$ then the subgraph of G induced by U is denoted by $G[U]$. For a vertex v, the set of vertices which are adjacent to v is called the *(open) neighborhood* of v and denoted by $N_G(v)$. The *closed neighborhood* of v is the set $N_G[v] = N_G(v) \cup \{v\}$. If $U \subseteq V(G)$ then $N_G[U] = \bigcup_{v \in U} N_G[v]$. The *distance* $\mathrm{dist}_G(u,v)$ between a pair of vertices u and v in a connected graph G is the number of edges on a shortest u,v-path in G. For a positive integer r, $N_G^r[v] = \{u \in V(G) \colon \mathrm{dist}_G(u,v) \le r\}$. Whenever there is no ambiguity we omit the subscripts.

The length of a shortest cycle in G is called the *girth* of G and denoted by $g(G)$. If G is an acyclic graph then $g(G) = +\infty$. We use $\Delta(G)$ for the maximum degree of a vertex in G. Let $C \subsetneq V(G)$, and $R = V(G) \setminus C$. We call the set R where the robber moves while trying to enter C the *robber-region*. A triple $[G;\, C, R]$ is called the *board* of the game. For convenience, we keep both sets C and R in our notation despite the fact that they define each other. Clearly, the game is fully specified by the number of cops c and the board. We call the set $\delta[G;\, C, R] = \{v \in C \colon N(v) \cap R \neq \emptyset\}$ the *boundary* of the board.

Since the game is played in alternating turns starting at turn 1, the cop-player moves his cops at odd turns, and robber-player moves the robber at even turns. Two consecutive turns $2 \cdot i - 1$ and $2 \cdot i$ are jointly referred to as a *round* i, $i \ge 1$.

A *state* of the game at *time* i is given by the positions of all cops and robbers on the board after $i - 1$ turns. A *strategy of a cop-player* (*strategy of a robber-player*) is a function $\mathcal{X}$ which, given the state of the game, determines the movements of the cops (the robber) in the current turn. If there are no cops (no robber) on the board, the function determines the initial positions of the cops (the robber).

The GUARDING problem is, given a board $[G;\, R, C]$, to compute the minimum number of cops that can guard the protected region C. We call this number the *guard number* of the board and denote it by $\mathbf{gn}(G; C, R)$. Despite the differences between the robber-first and cops-first games, some of the results established in [19] carry over to the cops-first game. In particular, the following claim holds (see also [8, 24, 26]).

Proposition 1 ([19, Proposition 1])**.** *There is an algorithm that given an integer $c \ge 1$ and a board $[G;\, C, R]$ with the n-vertex graph G determines whether c cops can guard C in time $\binom{|C|+c-1}{c}^2 \cdot |R|^2 \cdot n^{O(1)} = n^{O(c)}$.*

Thus for every fixed c, one can decide in polynomial time whether c cops can guard the protected region against the robber on a given graph G. The running time $n^{O(c)}$ cannot be improved to an FPT running time unless $FPT = W[2]$. We refer to the book of Downey and Fellows [16] for an introduction to parameterized complexity. A reduction from the DOMINATING SET problem yields the following proposition.

Proposition 2. [$\star$][1] *The* GUARDING *problem is NP-hard. The* GUARDING DECISION *problem parameterized by the number c of guards is* W[2]*-hard. Finally, there is a constant $\rho > 0$ such that there is no polynomial time algorithm that, for every instance, approximates the guard number within a multiplicative factor $\rho \log n$, unless* P = NP. *Both the hardness results and the inapproximability result hold even when the robber territory is an independent set or a path.*

3 Hardness of guarding

Fomin et al. proved in [19] that the robber-first variant of GUARDING problem is PSPACE-hard for *directed* graphs. For undirected graphs only NP-hardness was proved and PSPACE-hardness was left as an open question. In this section we prove that the cops-first game is PSPACE-hard both for undirected and directed graphs. It should be noted that the complexity analysis for Cops and Robbers games for undirected graphs is much more complicated than for directed graphs (see e.g. [24]).

[1] Proofs of results marked with [$\star$] have been moved to the appendix

Theorem 1. *The* GUARDING *problem is* PSPACE*-hard on undirected graphs.*

Proof. We reduce the PSPACE-complete QUANTIFIED BOOLEAN FORMULA IN CONJUNCTIVE NORMAL FORM (QBF) problem [22] to the decision variant of the GUARDING problem. For a set of Boolean variables $x_1, x_2, \ldots, x_n$ and a Boolean formula $F = C_1 \wedge C_2 \wedge \cdots \wedge C_m$, where C_j is a clause, the QBF problem asks whether the expression $\phi = Q_1 x_1 Q_2 x_2 \cdots Q_n x_n F$ is $true$, where for every i, Q_i is either $\forall$ or $\exists$.

Given a quantified Boolean formula ϕ, we construct an instance G of a guard game in several steps. We first construct a graph G' and show that if the robber strategy is restricted to some specific conditions, then ϕ is true if and only if the cop player can win on G' with a a specific number of cops. This part of the proof is described in Lemmata 1 and 2. Then we construct the graph G'' from G' by adding gadgets which force the robber to choose a particular vertex as starting vertex. Finally we construct the graph G from G' by adding gadgets that force the robber to follow the restricted strategy described in Lemmata 1 and 2. We prove that these gadgets indeed work as intended in Lemma 4.

Constructing G'. For every $Q_i x_i$ we introduce a gadget graph G_i. For $Q_i = \forall$, we define the graph $G_i(\forall)$ with vertex set $\{u_{i-1}, u_i, x_i, \overline{x}_i, y_i, \overline{y}_i, z_i, \overline{z}_i, a_i, \overline{a}_i, s_i, t_i\}$ and edge set $\{u_{i-1}y_i, y_i u_i, u_{i-1}\overline{y}_i, \overline{y}_i u_i, y_i a_i, a_i z_i, x_i z_i, \overline{y}_i \overline{a}_i, \overline{a}_i \overline{z}_i, \overline{x}_i \overline{z}_i, x_i s_i, x_i t_i, \overline{x}_i s_i, \overline{x}_i t_i\}$. Let $S_i = \{x_i, \overline{x}_i, z_i, \overline{z}_i, s_i, t_i\}$. For $Q_i = \exists$, we define $G_i(\exists)$ as the graph with vertex set $\{u_{i-1}, u_i, x_i, \overline{x}_i, y_i, z_i, a_i, s_i, t_i\}$ and edge set $\{u_{i-1}y_i, y_i u_i, y_i a_i, a_i z_i, x_i z_i, \overline{x}_i z_i, x_i s_i, x_i t_i, \overline{x}_i s_i, \overline{x}_i t_i\}$, and $S_i = \{x_i, \overline{x}_i, z_i, s_i, t_i\}$. The graphs $G_i(\forall)$ and $G_i(\exists)$ are shown in Figure 2. Observe that the vertex u_i appears both in the gadget graph G_i and in the gadget G_{i+1} for $i \in \{1, 2, \ldots, n-1\}$.

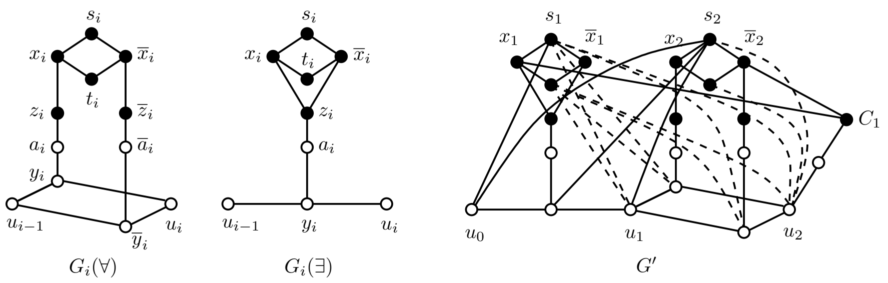

Figure 2: Graphs $G_i(\forall)$, $G_i(\exists)$ (vertices of S_i are shown by the black color) and G'

The graph G' also has vertices $C_1, C_2, \ldots, C_m$ corresponding to clauses. The vertex x_i is joined with C_j by an edge if C_j contains the literal x_i, and $\overline{x}_i$ is joined with C_j if C_j contains the literal $\overline{x}_i$. The vertex u_n is connected with all vertices $C_1, C_2, \ldots, C_m$ by paths of length two with middle vertices $w_1, w_2, \ldots, w_m$. For every $i \in \{1, 2, \ldots, n\}$, the vertex s_i is joined by edges with all vertices u_j, y_j and $\overline{y}_j$ for $0 \le j < i$, and the vertices s_i and t_i are connected by paths of length two with u_i and with all vertices u_j, y_j and $\overline{y}_j$ for $i < j \le n$. Let W be the set of middle vertices of these paths. This completes the construction of G'.

Let $C' = S_1 \cup S_2 \cup \cdots \cup S_n \cup \{C_1, C_2, \ldots, C_m\}$ be the cop-region of G', and $R' = V(G') \setminus C'$ be the robber-region. An example of G' for $\phi = \exists x_1 \forall x_2\; x_1 \vee \overline{x}_2$ is shown in Fig 2. The paths added in the last stage of the construction are shown by dashed lines and the vertices in W are not shown.

We proceed to prove several properties of G'.

Lemma 1. [$\star$] *If $\phi = false$, then the robber-player has a winning strategy on the board $[G'; C', R']$ against $c' = n$ cops.*

If the actions of the robber are restricted only to special strategies, then the condition $\phi = false$ is not only sufficient but also necessary for the robber to win.

Lemma 2. [$\star$] *Suppose that the robber can use only strategies with the following properties:*

- *he starts from u_0,*

- *he moves along edges $u_{i-1}y_i, y_iu_i, u_{i-1}\overline{y}_i, \overline{y}_iu_i$ only in the direction induced by this ordering, i.e. these edges are "directed" for him.*

Then if $\phi = true$, then $c' = n$ cops can win on $[G'; C', R']$.

Constructing G''. We now add gadgets to G' that "force" the robber-player to start in the vertex u_0. We take a path $bc_1c_2c_3p_0q_0p_1q_1p_2q_2 \dots p_{2n}q_{2n}$ and make each vertex u_i be adjacent with the vertices $p_{2i}, p_{2i+1}, \dots, p_{2n}$. Then we make the vertices y_i and $\overline{y}_i$ be adjacent to vertices $p_{2i-1}, p_{2i}, \dots, p_{2n}$. The vertex q_{2n} is adjacent to all vertices $z_i, \overline{z}_i, t_i$ and also to all vertices C_j. Denote the obtained graph by G'', and let $C'' = C' \cup \{c_1, c_2, c_3, p_0, q_0, p_1, q_1, p_2, q_2, \dots, p_{2n}, q_{2n}\}$, $R'' = R' \cup \{b\}$. See Fig 3, for the fragment of G''. This figure shows how the gadget is attached to G', where G' is taken from the example in Fig 2.

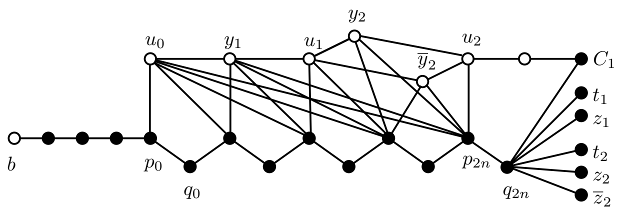

Figure 3: Construction of G''

Properties of G'' are summarized in the next lemma.

Lemma 3. [$\star$] *Let $c'' = 3n + 2$.*

- *If $\phi = false$, then the robber can win on $[G''; C'', R'']$ against c'' cops;*
- *If the starting vertex r of the robber is not u_0, then c'' cops win;*
- *If the robber can move along edges $u_{i-1}y_i, y_iu_i, u_{i-1}\overline{y}_i, \overline{y}_iu_i$ only from the first vertex to the next and $\phi = true$, then c'' cops win.*

Constructing G. Finally we add a gadget which makes it pointless for the robber to move on the edges $u_{i-1}y_i, y_iu_i, u_{i-1}\overline{y}_i, \overline{y}_iu_i$ in the "wrong" direction. We introduce the path $P = der_0r_1 \dots r_{2n+1}$, and two vertices f_1 and f_2. The vertices f_1 and f_2 are made adjacent to vertices $r_0, r_1, \dots, r_{2n+1}$, and they are joined with all vertices $u_i, y_i, \overline{y_i}$ by paths of length two. Denote by W_1 the set of middle vertices of paths with the endpoint f_1, and by W_2 the set of middle vertices of paths with the endpoint f_2. Every vertex r_k is made adjacent to all vertices $z_i, \overline{z_i}$ for $1 \leq i \leq \frac{k}{2}$. Vertex r_{2n+1} is adjacent to $C_1, C_2, \dots, C_m$. Denote the obtained graph G, and define $C = C'' \cup \{e, r_0, r_1, \dots, r_{2n+1}, f_1, f_2\}$, $R = V(G) \setminus C$. See Figure 4 for the fragment of G. In this figure, the gadget is attached to G'' depicted in Fig 3. The paths added in the last stage of the construction are shown by dashed lines and the vertices of W_1 and W_2 are not shown.

Lemma 4. [$\star$] *The robber has a winning strategy on $[G; C, R]$ against $c = 3n + 3$ cops if and only if $\phi = false$.*

The size of the graph G is bounded by a polynomial of n and m, and therefore, the proof of Lemma 4 completes the proof of the theorem. □

The statement of Theorem 1 also holds for directed graphs since we can model an edge with two arcs, one going in each direction. Moreover, by using a simplified variant of our reduction, it can be proved that the GUARDING problem is PSPACE-hard even on directed acyclic graphs.

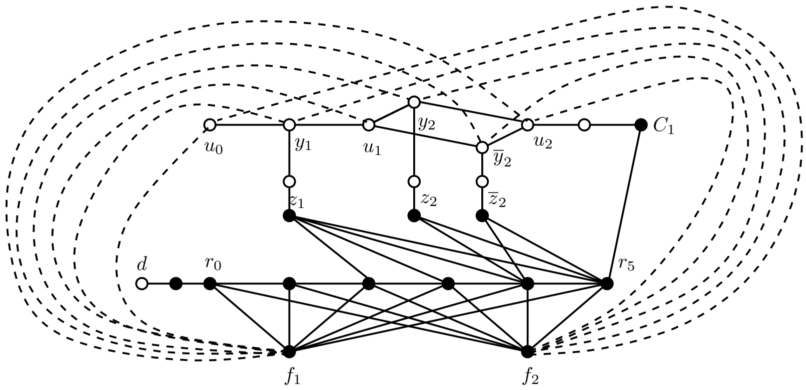

Figure 4: Construction of G

4 One-step guarding

4.1 The One-step guard number

In any cop-winning strategy, when the robber occupies some vertex $u \in R$, the cops should prevent him from entering C by blocking all vertices of $C \cap N(u)$. Since the robber makes his first move after the cops have chosen their initial positions, the cops have to start from an initial position such that for every vertex $u \in R$ they can occupy all vertices of $C \cap N(u)$ in one step. Thus it is not unreasonable that the number of cops needed to protect C from a robber that is only allowed to make one move after the initial step approximates the guard number of the board. Consider the variant of the game, where the robber is allowed to make only one move after the placement step. We call this variant of the game the *one-step* game. Then the minimum number of cops sufficient to guard the graph in this game is called the *one-step guard number*, and we denote the one-step guard number for the board $[G;\, C, R]$ by $\mathbf{gn}_1(G; C, R)$. We call the problem of computing the *one-step guard number* of a graph by the ONE-STEP GUARDING problem. In the ONE-STEP GUARDING problem, a strategy for the cop-player on the board $[G;\, C, R]$ is defined as a pair $\mathcal{S} = (s, \mathcal{F})$ where

- s is a mapping assigning to every vertex v of C a non-negative integer $s(v)$ — the number of cops in v.
- $\mathcal{F} = \{f_u\}_{u \in R}$ is a family of functions $f_u \colon C \cap N(u) \to C$ defining moves of cops if the robber occupies u (a cop moves to $w \in C \cap N(u)$ from $f_u(w)$), such that for every $w \in C \cap N(u)$, $f_u(w) \in N[w]$, and for every $v \in C$, $|\{w \in C \cap N(u) \colon f_u(w) = v\}| \le s(v)$.

If $X \subseteq C$, then $s(X) = \sum_{v \in X} s(v)$. We say that $\mathcal{S}$ is a winning strategy for c cops if $s(C) \le c$. The simple but useful property of the one-step guard number is that it depends only on the local structure of the border neighborhood. We formalize this property in the following proposition, whose proof follows directly from the definition of one-step guarding.

Proposition 3. *For every board* $[G;\, C, R]$, $\mathbf{gn}_1(G; C, R) = \mathbf{gn}_1(G'; C', R')$ *for* $G' = G[N_G[\delta[G;\, C, R]]]$, $C' = C \cap N_G[\delta[G;\, C, R]]$ *and* $R' = R \cap N_G[\delta[G;\, C, R]]$.

The one-step guard number gives the following approximation of the guard number.

Theorem 2. *For any board* $[G;\, C, R]$, $\mathbf{gn}_1(G; C, R) \le \mathbf{gn}(G; C, R) \le 2 \cdot \mathbf{gn}_1(G; C, R)$.

Proof. The lower bound follows directly from the definitions of both games. To prove the upper bound let us assume that $\mathcal{S} = (s, \mathcal{F})$ is a winning strategy for $\mathbf{gn}_1(G; C, R)$ cops in the one-step game.We put $2s(v)$ cops on every vertex $v \in C$ and divide them into two equal size teams. Then the cops perform the following actions. When the robber moves to some vertex u, the cops from the first team move to all vertices of $C \cap N(u)$ according to the mapping f_u. When the robber moves to another vertex w the cops from the first team return to their original positions, and the cops from the second team move to all vertices of $C \cap N(w)$. Then the second team returns and the first team moves to guard C, and so on. Clearly, this is a winning strategy for $2 \cdot \mathbf{gn}_1(G; C, R)$ cops in the original guard game. $\square$

A tightness of the upper bound can be seen from the following example. Let G be a graph with vertices x, y, z, such that the vertices x and y are adjacent. The vertices x and z, and the vertices y and z are joined by k paths of length two. Let $R = \{x, y\}$, and $C = V(G) \setminus R$. It can be easily shown that $\mathbf{gn}_1(G; C, R) = k$ and $\mathbf{gn}(G; C, R) = 2k$. We now show that the lower bound is tight for a large collection of boards.

Theorem 3. *Let $[G;\, C, R]$ be a board such that for every cycle C_5 of length 5 in G, $|E(C_5) \cap E(G[R])| \neq 1$. Then $\mathbf{gn}_1(G; C, R) = \mathbf{gn}(G; C, R)$.*

Proof. Suppose every cycle of length 5 either has more than one edge in $G[R]$, or has no edges at all. Since $\mathbf{gn}_1(G; C, R) \leq \mathbf{gn}(G; C, R)$ always holds, it is sufficient to prove that $\mathbf{gn}_1(G; C, R) \geq \mathbf{gn}(G; C, R)$. Let $\mathcal{S} = (s, \mathcal{F})$ be a winning strategy for $\mathbf{gn}_1(G; C, R)$ cops in the one-step game. We describe the strategy for $\mathbf{gn}_1(G; C, R)$ cops in the general guard game as follows. We put $s(v)$ cops on every vertex $v \in C$. When the robber moves to some vertex u, the cops move to all vertices of $C \cap N(u)$ according to f_u. When the robber moves to another vertex w from u, the cops which moved in the previous round, return to their original positions and other cops move to all vertices of $C \cap N(w)$ according to f_w. If $f_u(x) = f_w(x)$ for some $x \in C \cap N(u) \cap N(w)$, then the cop remains in x. The only possible situation in which the cops are not able to move as described above is if there are vertices $x \in C \cap N(u)$ and $y \in C \cap N(w)$, $x \neq y$, for which $f_u(x) = f_w(y)$, $f_u(x) \neq x$ and $f_w(y) \neq y$. This can happen only if C_5 is a subgraph of G with exactly one edge uw in $G[R]$. Now we can assume that $u := w$ and repeat the above strategy. $\square$

Since the set R in the proof of Proposition 2 is an independent set, it immediately follows from Propositions 2 and 3 that the computation of the one step guard number is difficult.

Corollary 1. *The decision version of the* ONE-STEP GUARDING *problem is* NP*-complete and it remains* NP*-complete for planar graphs. Moreover, the parameterized version of the problem with k being a parameter is* W[2]*-hard. Also, there is a constant $\rho > 0$ such that there is no polynomial time algorithm that, for every instance, approximate the border dominating number within a multiplicative factor $\rho \log n$, unless* P = NP.

Despite the algorithmic lower bounds in Corollary 1, it is sometimes possible to use the one-step guard number for an approximation of the guard number.

Let us consider a generalization of the DOMINATING SET problem called BLACK AND WHITE DOMINATING SET problem (see e.g. [3]). The input is a *black and white* graph, which simply means that the vertex set of the graph G has been partitioned into two disjoint sets B and W of black and white vertices. Given a black and white graph G, the problem is to find a *dominating* set $X \subset V(G)$ of the minimum cardinality which dominates B, i.e. such that for each vertex $v \in B$, $N_G[v] \cap X \neq \emptyset$. We call the cardinality of such a set the *black and white domination number* and denote it by $\gamma(G; B, W)$. Observe for any cop-winning strategy the set of vertices occupied by the cops in the beginning of the game has to dominate the boundary $\delta[G;\, C, R]$. This yields the following proposition about the relationship between black and white domination and one-step graph guarding.

Proposition 4. *For any board $[G;\, C, R]$, $\gamma(G[C]; \delta[G;\, C, R], C \setminus \delta[G;\, C, R]) \leq \mathbf{gn}_1(G; C, R)$.*

These two parameters can be arbitrarily far apart. Consider the graph G constructed from two vertices u and v by joining them by k paths of length two with middle vertices $w_1, \ldots, w_k$, and let $C = \{v, w_1, \ldots, w_k\}$. Obviously, $\mathbf{gn}_1(G; C, R) = k$ and $\gamma(G[C]; \delta[G;\, C, R], C \setminus \delta[G;\, C, R]) = 1$. Still there are cases when these parameters coincide.

Proposition 5. [$\star$] *Let $[G;\, C, R]$ be a board such that $g(G) \geq 5$. Then $\gamma(G[C]; \delta[G;\, C, R], C \setminus \delta[G;\, C, R]) = \mathbf{gn}_1(G; C, R)$.*

Combining Theorem 3 and Proposition 5, we obtain the next corollary.

Corollary 2. *Let* $[G;\, C, R]$ *be a board such that* $g(G) \geq 6$. *Then* $\gamma(G[C]; \delta[G;\, C, R], C \setminus \delta[G;\, C, R]) = \mathbf{gn}_1(G; C, R) = \mathbf{gn}(G; C, R)$.

It is known [30] that the parameterized variant of the BLACK AND WHITE DOMINATING SET problem with the cardinality of dominating set being the parameter is FPT for graphs of girth at least 5. Together with Theorem 3 this yields the following corollary.

Corollary 3. *The* (ONE-STEP) GUARDING *and* GUARDING *problems are* FPT *when parameterized by the number of cops for boards* $[G;\, C, R]$ *with* $g(G) \geq 6$.

4.2 One-step guarding for sparse graphs

In this section we consider the ONE-STEP GUARDING problem in graphs of bounded treewidth. Due to lack of space the definition of treewidth, tree-decomposition and nice tree-decomposition has been moved to the appendix. It is well known that many problems, which are difficult for generals graphs, can be solved in polynomial time in graphs of bounded treewidth. We show here that this is also the case for the computation of the one-step guard number. We construct a dynamic programming algorithm for this problem.

Theorem 4. $[\star]$ *Let* G *be an* n *vertex graph given with its tree decomposition of width* t. *Then* $\mathbf{gn}_1(G; C, R)$ *can be computed in time* $h(t)n^{O(t^2)}$, *where* h *is some function of* t.

Note that this algorithm is polynomial if the treewidth if fixed, but it is not an FPT algorithm when t is the parameter. In what follows, we show that (up to widely believed assumption that $\mathrm{FPT} \neq \mathrm{W}[1]$) the ONE-STEP GUARDING problem parameterized by the treewidth of the input graph is not FPT.

Theorem 5. *The* ONE-STEP GUARDING *problem is* $\mathrm{W}[1]$*-hard when parameterized by the treewidth of the input graph.*

Proof. We reduce from the CAPACITATED DOMINATING SET problem. A *capacitated graph* is a pair (G, c) where G is a graph and $c\colon V(G) \to \mathbb{N}$ is a *capacity* function such that $1 \leq c(v) \leq \deg v$ for every vertex $v \in V(G)$. A set $S \subset V(G)$ is called a *capacitated dominating set* if there is a *domination mapping* $g\colon V(G) \setminus S \to S$ which maps every vertex in $(V(G) \setminus S)$ to one of its neighbors such that the total number of vertices mapped by g to any vertex $v \in S$ does not exceed its capacity $c(v)$. The CAPACITATED DOMINATING SET problem is defined as follows. Given a capacitated graph (G, c) and a positive integer k, determine whether there exists a capacitated dominating set S for G containing no more than k vertices. It was proved by Dom et al. [15] that this problem is $W[1]$-hard when parameterized by treewidth and k.

We start with descriptions of auxiliary gadgets. For a positive integer r, we construct the graph $G(r)$ as follows. Two vertices u and v are introduced and joined by r paths of length three. Denote by ux_iy_iv the i-th path. Then the vertex w is added and joined by edges with vertices $y_1, y_2, \ldots, y_r$, and for every vertex x_i, a leaf z_i is included and joined with x_i. The example of such a graph is shown in Fig 5. Let $R(G(r)) = \{w, z_1, z_2, \ldots, z_r\}$ and $C(G(r)) = V(G(r)) \setminus R(G(r))$.

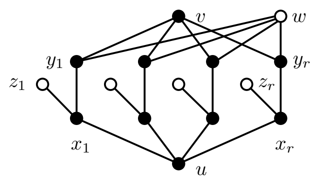

Figure 5: Graph $G(r)$

This graph has the following properties.

Lemma 5. *Suppose that* $\mathcal{S} = (s, \mathcal{F})$ *is a strategy for the cop-player for the board* $[G(r);\, C(G(r)), C(G(r))]$. *Then*

- *$s(C(G(r)) \setminus \{u\}) \geq r$;*
- *if $s(C(G(r))) = r$, then $s(u) = 0$ and $s(v) = 0$.*

Also, let

$$s_1(t) = \begin{cases} r & \text{if } t = v, \\ 1 & \text{if } t = u, \\ 0 & \text{if } t \neq u, v; \end{cases} \quad \text{and } s_2(t) = \begin{cases} 1 & \text{if } t = y_i, \\ 0 & \text{if } t \neq y_i. \end{cases}$$

Then there are cop strategies $\mathcal{S}_1 = (s_1, \mathcal{F}_1)$ and $\mathcal{S}_2 = (s_2, \mathcal{F}_2)$ for $[G(r);\, C(G(r)), C(G(r))]$.

Proof. The first claim follows immediately from the observation that for $f_w \in \mathcal{F}$, $f_w(y_i) \in \{x_i, y_i, v\}$, and therefore, $\sum_{i=1}^{r} s(x_i) + \sum_{i=1}^{r} s(y_i) + s(v) \geq r$.

Since $s(C(G(r)) \setminus \{u\}) \geq r$, we have that if $s(C(G(k))) = r$, then $s(u) = 0$. Suppose that $s(v) \neq 0$. Then for some $i \in \{1, 2, \ldots, r\}$, $s(x_i) = s(y_i) = 0$. But $f_{z_i}(x_i) \in \{x_i, y_i, u\}$, which is a contradiction. The last claim is true because we can define $f_w(y_i) = v$ and $f_{z_i}(x_i) = u$ in $\mathcal{F}_1$, and $f_w(y_i) = x_i$ and $f_{z_i}(x_i) = x_i$ in $\mathcal{F}_2$. □

Now we are ready to describe our reduction. Let (G, c) be a capacitated graph with the vertex set $\{a_1, a_2, \ldots, a_n\}$, and let k be a positive integer. For every $i \in \{1, 2, \ldots, n\}$, we introduce a copy of $G(c(a_i))$. Denote this graph by G_i, and denote by u_i and v_i vertices u and v of G_i. For every edge $a_i a_j$ of G, a pair of edges $u_i v_j$ and $u_j v_i$ is added. Then $2k$ vertices $b_1, b_2, \ldots, b_k$ and $d_1, d_2, \ldots, d_k$ are included, and all vertices b_i and d_i are joined by edges with $u_1, u_2, \ldots, u_n$. Now a vertex p is added and joined with $u_1, u_2, \ldots, u_n$. And, finally, vertices q_1 and q_2 are introduced, q_1 is joined with $b_1, b_2, \ldots, b_k$ by edges, and q_2 is joined with $d_1, d_2, \ldots, d_k$. Denote the obtained graph by H, and let $C = (\bigcup_{i=1}^{n} C(G_i)) \cup \{b_1, b_2, \ldots, b_k\} \cup \{d_1, d_2, \ldots, d_k\}$ and $R = (\bigcup_{i=1}^{n} R(G_i)) \cup \{p, q\}$. Let also $r = \sum_{i=1}^{n} c(a_i) + k$.

Lemma 6. [$\star$] *Graph (G, c) has a capacitated dominating set of a size at most k if and only if $\mathbf{gn}_1(H; C, R) \leq r$.*

Now we prove that when the treewidth of G is bounded the treewidth of H is bounded as well.

Lemma 7. [$\star$] $\mathbf{tw}(H) \leq 2 \cdot \mathbf{tw}(G) + 2k + 4$.

The CAPACITATED DOMINATING SET problem is W[1]-hard if parameterized both by the size of the capacitated dominating set and the treewidth, and this completes the proof of the theorem. □

Since the set R in the proof of the theorem is independent, by Proposition 3, we have the following corollary.

Corollary 4. *The parameterized version of the* GUARDING *problem with the treewidth of the input graph being a parameter is* W[1]*-hard.*

In the following theorem we show that with some additional restrictions on graphs the ONE-STEP GUARDING problem become fixed parameter tractable.

Theorem 6. [$\star$] *For any positive integers t and d, $\mathbf{gn}_1(G; C, R)$ can be computed in linear time for boards $[G;\, C, R]$, with $\mathbf{tw}(G) \leq t$ and $\Delta(G) \leq d$.*

4.3 One-step guarding for apex-minor-free graphs

Our results for graphs of bounded treewidth can be used for approximation of the one-step guard number for some graph classes.

It is said that a graph class $\mathcal{G}$ has *bounded local treewidth with bounding function f* if there is a function $f : \mathbb{N} \to \mathbb{N}$ such that for every graph $G \in \mathcal{G}$, every $v \in V(G)$, and every positive integer r it holds that $\mathbf{tw}(G[N^r[v]]) \leq f(r)$. An *apex graph* is a graph obtained from a planar graph G by adding a vertex and making it adjacent to some vertices of G. A graph class is *apex-minor-free* if it does not contain any graph with some fixed apex graph as a minor. For example, planar graphs (and bounded-genus graphs) are apex-minor-free graphs.

Eppstein [17, 18] characterized all minor-closed graph classes that have bounded local treewidth. It was proved that they are exactly apex-minor-free graphs. These results were improved by Demaine and Hajiaghayi [14]. They proved that all apex-minor-free graphs have linear local treewidth. We show that there is a polynomial time approximation scheme (PTAS) on the class of apex-minor-free graphs for the computation of the one-step guard number. To do this we use the well known technique for solving NP-hard problems on planar graphs proposed by Baker [7] and generalized by Eppstein [17, 18] (see also [14]) to minor-closed graph classes with bounded local treewidth. We start with a simple observation.

Lemma 8. [$\star$] *Let $[G_1;\, C_1, R_2]$ and $[G_2;\, C_2, R_2]$ be two boards such that $C_1 \cap R_2 = C_2 \cap R_1 = \emptyset$. Then $\mathbf{gn}_1(G; C, R) \leq \mathbf{gn}_1(G_1; C_1, R_1) + \mathbf{gn}_1(G_2, C_2, R_2)$, where $G = G_1 \cup G_2$, $C = C_1 \cup C_2$ and $R = R_1 \cup R_2$.*

Let u be a vertex of a graph G. For $i \geq 0$ we denote by L_i the i-th level of breadth first search from u, i.e. the set of vertices at distance i from u. We call the partition of the vertex set $V(G)$ $\mathcal{L}(G, u) = \{L_0, L_1, \ldots, L_r\}$ *breadth first search (BFS) decomposition* of G. We assume for convenience that for a BFS decomposition $\mathcal{L}$, (G, u) $L_i = \emptyset$ whenever $i < 0$ or $i > r$, The BFS decomposition can be constructed using a breadth first search in a linear time.

Let $[G;\, C, R]$ be a board, and let G be a graph with BFS decomposition $\mathcal{L}(G, u) = (L_0, L_1, \ldots, L_r)$, and t be a positive integer. Suppose that $i \leq j$ are integers. For $i \leq j$, we define $G_{ij} = G[\bigcup_{p=i}^{j} L_p]$. For all $i \leq j$, we set $C_{i,j} = C \cap G_{i-2,j+2}$, $R_{i,j} = R \cap G_{i,j}$ and $F_{i,j} = G[C_{i,j} \cup R_{i,j}]$.

The following result is due to Demaine and Hajiaghayi [14] (see also the paper by Eppstein [18]),

Lemma 9 ([14])**.** *Let G be an apex-minor-free graph. Then $\mathbf{tw}(F_{ij}) = O(j - i)$.*

Now we are ready to describe our algorithm. Let $k \geq 4$ be an integer. For a given board $[G;\, C, R]$ for an apex-minor-free graph G, the BFS decomposition $\mathcal{L}(G, u) = (L_0, L_1, \ldots, L_r)$ is constructed for some vertex u.

If $r \leq k$ then $\mathbf{gn}_1(G; C, R)$ is computed directly. We use the fact that $\mathbf{tw}(G) = O(k)$ and, for example, use Bodlaender's Algorithm [9] to construct in linear time a suitable tree decomposition of G. Then, by Theorem 4, $\mathbf{gn}_1(G; C, R)$ can be computed in a polynomial time.

Suppose now that $r > k$. Let $F_i = F_{i,i+k-1}$, $C_i = C_{i,i+k-1}$ and $R_i = R_{i,i+k-1}$. For $i = 1, \ldots, k$, we construct boards $[F_{i+(j-1)\cdot k};\, C_{i+(j-1)\cdot k}, R_{i+(j-1)\cdot k}]$ for $0 \leq j \leq p = \lceil \frac{r-i+1}{k} \rceil + 1$, and compute

$$c_i = \sum_{j=0}^{p} \mathbf{gn}_1(F_{i+(j-1)\cdot k}; C_{i+(j-1)\cdot k}, R_{i+(j-1)\cdot k}).$$

We approximate $\mathbf{gn}_1(G; C, R)$ by the value $\mathbf{gn}'_1(G; C, R) = \min\{c_i \colon i \in \{1, \ldots, k\}\}$.

The following lemma gives properties of the algorithm.

Lemma 10. *For any board $[G;\, C, R]$ for an apex-minor-free graph G and for each fixed positive integer k,*

1. *$\mathbf{gn}'_1(G; C, R)$ can be computed in a polynomial time.*

2. *$\mathbf{gn}_1(G; C, R) \leq \mathbf{gn}'_1(G; C, R) \leq (1 + \frac{4}{k}) \cdot \mathbf{gn}_1(G; C, R)$.*

Proof. We use the fact than $\bigcup_{j=1}^{p} F_{i+(j-1)\cdot k} = G$, $\bigcup_{j=1}^{p} C_{i+(j-1)\cdot k} = C$ and $\bigcup_{j=1}^{p} R_{i+(j-1)\cdot k} = R$.

The first claim of the lemma follows immediately from Theorem 4 and Lemma 9. The inequality $\mathbf{gn}_1(G; C, R) \leq \mathbf{gn}'_1(G; C, R)$ is an easy corollary of Lemma 8. So it remained to prove the inequality $\mathbf{gn}'_1(G; C, R) \leq (1 + \frac{4}{k}) \cdot \mathbf{gn}_1(G; C, R)$.

Let $\mathcal{S} = (s, \mathcal{F})$ be a strategy for $\mathbf{gn}(G; C, R)$ cops on the board $[G;\, C, R]$. Consider the strategy $\mathcal{S}_i = (s_i, \mathcal{F}_i)$ for $[F_i;\, C_i, R_i]$ where $s_i(v) = s(v)$ for $v \in C_i$ and $\mathcal{F}_i = \{f_u \in \mathcal{F} \colon u \in R_i\}$. Since for any $u \in R_i$ and each $v \in N_G[u] \cap C$, $N_G[v] \cap C \subseteq C_i$, $\mathcal{S}_i$ is a valid winning strategy for $s(C_i)$ cops and hence $\mathbf{gn}_1(F_i; C_i, R_i) \leq s(C_i)$. Observe that for consecutive sets $C_{i+(j-1)\cdot k}$ and $C_{i+j\cdot k}$, $C_{i+(j-1)\cdot k} \cap C_{i+j\cdot k} \subseteq (L_{i+j\cdot k-2} \cup L_{i+j\cdot k-1} \cup L_{i+j\cdot k} \cup L_{i+j\cdot k+1}) \cap C$. Since $\mathbf{gn}'_1(G; C, R) = \min\{c_i \colon i \in \{1, \ldots, k\}\}$, $\mathbf{gn}'_1(G; C, R) \leq s(C) + \frac{4}{k} \cdot s(C) = (1 + \frac{4}{k}) \cdot \mathbf{gn}_1(G; C, R)$. □

Finally, we have the following claim.

Theorem 7. *The problem of computation of the one-step guard number admits a PTAS for classes of apex-minor-free graphs and hence, for planar graphs and for graphs with bounded genus).*

Using Theorem 2, we get an approximation for the guard number.

Corollary 5. *For any $\varepsilon > 0$, the guard number can be approximated within the factor $2 + \epsilon$ in a polynomial time for apex-minor-free graph classes.*

Notice that Corollary 5 together with Proposition 3 yields a PTAS for the guard number of a restricted class of apex-minor-free graphs. This class includes, but is not limited to the apex-minor-free graphs with girth at least 6. For some special cases it is possible to get better results for planar graphs.

Theorem 8. [$\star$] *Let $[G;\, C, R]$ be a board such that G is a planar graph which is embedded in such a way that all vertices of $R \cap N[C]$ lay on the boundary of the external face of $G[N[C]]$. Then $\mathbf{gn}_1(G; C, R)$ can be computed polynomially.*

This theorem means that in this case the guard number can be approximated polynomially within the factor 2. Moreover, for some cases (see Proposition 3) the guard number itself can be computed polynomially. For example, when G is bipartite. Note also that it is possible to give a "symmetric" sufficient conditions for the board.

References

[1] M. AIGNER AND M. FROMME, *A game of cops and robbers*, Discrete Appl. Math., 8 (1984), pp. 1–11.

[2] J. ALBER, H. L. BODLAENDER, H. FERNAU, T. KLOKS, AND R. NIEDERMEIER, *Fixed parameter algorithms for dominating set and related problems on planar graphs*, Algorithmica, 33 (2002), pp. 461–493.

[3] J. ALBER, H. FAN, M. R. FELLOWS, H. FERNAU, R. NIEDERMEIER, F. ROSAMOND, AND U. STEGE, *A refined search tree technique for dominating set on planar graphs*, J. Comput. System Sci., 71 (2005), pp. 385–405.

[4] B. ALSPACH, *Searching and sweeping graphs: a brief survey*, Matematiche (Catania), 59 (2006), pp. 5–37.

[5] M. ANDERSON, C. BARRIENTOS, R. C. BRIGHAM, J. R. CARRINGTON, R. P. VITRAY, AND J. YELLEN, *Maximum-demand graphs for eternal security*, J. Combin. Math. Combin. Comput., 61 (2007), pp. 111–127.

[6] S. ARNBORG, J. LAGERGREN, AND D. SEESE, *Easy problems for tree-decomposable graphs*, J. Algorithms, 12 (1991), pp. 308–340.

[7] B. S. BAKER, *Approximation algorithms for NP-complete problems on planar graphs*, J. Assoc. Comput. Mach., 41 (1994), pp. 153–180.

[8] A. BERARDUCCI AND B. INTRIGILA, *On the cop number of a graph*, Adv. in Appl. Math., 14 (1993), pp. 389–403.

[9] H. L. BODLAENDER, *A linear-time algorithm for finding tree-decompositions of small treewidth*, SIAM J. Comput., 25 (1996), pp. 1305–1317.

[10] ———, *Treewidth: Algorithmoc techniques and results*, in MFCS, I. Prívara and P. Ruzicka, eds., vol. 1295 of Lecture Notes in Computer Science, Springer, 1997, pp. 19–36.

[11] H. L. BODLAENDER, *A partial k-arboretum of graphs with bounded treewidth*, Theoretical Computer Science, 209 (1998), pp. 1–45.

[12] A. P. BURGER, E. J. COCKAYNE, W. R. GRÜNDLINGH, C. M. MYNHARDT, J. H. VAN VUUREN, AND W. WINTERBACH, *Finite order domination in graphs*, J. Combin. Math. Combin. Comput., 49 (2004), pp. 159–175.

[13] A. P. BURGER, E. J. COCKAYNE, W. R. GRÜNDLINGH, C. M. MYNHARDT, J. H. VAN VUUREN, AND W. WINTERBACH, *Infinite order domination in graphs*, J. Combin. Math. Combin. Comput., 50 (2004), pp. 179–194.

[14] E. D. DEMAINE AND M. T. HAJIAGHAYI, *Equivalence of local treewidth and linear local treewidth and its algorithmic applications*, in SODA, J. I. Munro, ed., SIAM, 2004, pp. 840–849.

[15] M. DOM, D. LOKSHTANOV, S. SAURABH, AND Y. VILLANGER, *Capacitated domination and covering: A parameterized perspective*, IWPEC 2008, to appear.

[16] R. G. DOWNEY AND M. R. FELLOWS, *Parameterized complexity*, Springer-Verlag, New York, 1999.

[17] D. EPPSTEIN, *Subgraph isomorphism in planar graphs and related problems.*, in SODA, 1995, pp. 632–640.

[18] ———, *Diameter and treewidth in minor-closed graph families.*, Algorithmica, 27 (2000), pp. 275–291.

[19] F. V. FOMIN, P. A. GOLOVACH, A. HALL, M. MIHALÁK, E. VICARI, AND P. WIDMAYER, *How to guard a graph?*, in ISAAC, S.-H. Hong, H. Nagamochi, and T. Fukunaga, eds., vol. 5369 of Lecture Notes in Computer Science, Springer, 2008, pp. 318–329.

[20] F. V. FOMIN, P. A. GOLOVACH, AND KRATOCHVÍL, *Complexity for cops and robbers games*, manuscript, (2007).

[21] F. V. FOMIN AND D. M. THILIKOS, *An annotated bibliography on guaranteed graph searching*, Theor. Comput. Sci., 399 (2008), pp. 236–245.

[22] M. R. GAREY AND D. S. JOHNSON, *Computers and intractability*, W. H. Freeman and Co., San Francisco, Calif., 1979. A guide to the theory of NP-completeness, A Series of Books in the Mathematical Sciences.

[23] W. GODDARD, S. M. HEDETNIEMI, AND S. T. HEDETNIEMI, *Eternal security in graphs*, J. Combin. Math. Combin. Comput., 52 (2005), pp. 169–180.

[24] A. S. GOLDSTEIN AND E. M. REINGOLD, *The complexity of pursuit on a graph*, Theoret. Comput. Sci., 143 (1995), pp. 93–112.

[25] J. L. GOLDWASSER AND W. F. KLOSTERMEYER, *Tight bounds for eternal dominating sets in graphs*, Discrete Math., 308 (2008), pp. 2589–2593.

[26] G. HAHN AND G. MACGILLIVRAY, *A note on k-cop, l-robber games on graphs*, Discrete Math., 306 (2006), pp. 2492–2497.

[27] W. F. KLOSTERMEYER, *Complexity of eternal security*, J. Combin. Math. Combin. Comput., 61 (2007), pp. 135–140.

[28] W. F. KLOSTERMEYER AND G. MACGILLIVRAY, *Eternally secure sets, independence sets and cliques*, AKCE Int. J. Graphs Comb., 2 (2005), pp. 119–122.

[29] ———, *Eternal security in graphs of fixed independence number*, J. Combin. Math. Combin. Comput., 63 (2007), pp. 97–101.

[30] V. RAMAN AND S. SAURABH, *Short cycles make W-hard problems hard: FPT algorithms for W-hard problems in graphs with no short cycles*, Algorithmica, 52 (2008), pp. 203–225.

[31] R. RAZ AND S. SAFRA, *A sub-constant error-probability low-degree test, and a sub-constant error-probability PCP characterization of NP*, in STOC, 1997, pp. 475–484.

Appendix - Omitted Proofs

4.4 Proof of Proposition 2

Proof. We reduce from the DOMINATING SET problem. This problem asks about the existence of a set $S \subset V(G)$ of the size at most k such that $N[S] = V(G)$. It is well known ([16]) that this problem W[2]-hard when k is the parameter. For a graph G, we construct the graph G' by adding one leaf to each vertex of G. Let $C = V(G)$ and $R = V(G') \setminus V(G)$. It is easy to see that k cops guard the board $[G'; C, R]$ if and only if there is a dominating set of the size at most k in G. We combine this reduction and the non-approximability of the MINIMUM DOMINATING SET problem [31] to arrive at the inapproximability of the GUARDING problem. This proves the statement of the proposition for the case when the robbers territory is an independent set. To prove the statement for a path, one should connect the added leaves to form a path, and subdivide each edge of this path by two vertices. □

Proof of Theorem 1

Proof. Suppose that $\phi = false$. We describe a winning strategy for the robber-player. Independently of the initial positioning of the cops, the robber places himself on u_0. After this, the cops must respond by occupying $s_1, s_2, \ldots, s_n$, because otherwise the robber wins in the next move. Now, the robber starts moving towards the vertex u_n along some path in $G'[R']$. Every time the robber is placed on a vertex y_i of $G_i(\forall)$, there should be a cop responding to this move by moving to x_i from s_i, and if the robber occupies $\overline{y}_i$, then some cop has to move to $\overline{x}_i$. Otherwise the robber can move onto z_i or $\overline{z}_i$ moving from y_i or $\overline{y_i}$ along a path of length two. Note that the cop standing on s_i cannot leave s_i before the robber enters y_i or $\overline{y}_i$, because otherwise the robber could move to s_i and win. Thus, the cops are "forced" to occupy vertices that correspond to literals. Similarly, if the robber occupies the vertex y_i in $G_i(\exists)$, then a cop is forced to move from s_i to x_i or $\overline{x}_i$, and this cop can choose which vertex out of x_i and $\overline{x}_i$ to occupy. In both cases, a cop can not leave from the vertices x_i or $\overline{x}_i$ after the robber leaves y_i or $\overline{y}_i$, since otherwise the robber can move to s_i or to t_i along the path of length two from his current position. Since $\phi = false$, we have that the robber can choose between y_i and $\overline{y}_i$ in the gadgets $G_i(\forall)$ such that no matter how the cop player chooses to place the cops on x_i or $\overline{x}_i$ in the gadgets $G_i(\exists)$, when the robber arrives at u_n at least one vertex C_j has no cops on a vertex adjacent to it. Then the robber moves to this vertex along the edges $u_n w_j$, $w_j C_j$, and enters the cops territory. □

Proof of Lemma 2

Proof. Assume that $\phi = true$. We describe a winning strategy for the cop-player. The cops start by occupying vertices $s_1, s_2, \ldots, s_n$. If at any point during the game the robber moves to a vertex y_i from u_{i-1} of $G_i(\forall)$, then the cop occupying s_i moves to x_i and it the corresponding variable x_i is set to $true$. If the robber moves to $\overline{y}_i$, then the cop moves to $\overline{x}_i$ and $x_i = false$. It means that for a quantified variable $\forall x_i$, the robber chooses the value of x_i. If the robber moves to y_i of $G_i(\exists)$ from u_{i-1}, then the cops reply by moving a cop from s_i to x_i or $\overline{x}_i$, and this represents the value of the variable x_i. So for a quantified variable $\exists x_i$, the cops choose the value of x_i. If the robber moves from y_i to a_i in $G_i(\forall)$, then a cop moves from x_i to z_i, and if the robber moves back to y_i the cop returns to x_i. The cops use the same strategy for the case when the robber moves from $\overline{y}_i$ to $\overline{a}_i$ in $G_i(\exists)$. If the robber tries to move towards s_i or t_i along some path of length two, then a cop moves from x_i or $\overline{x}_i$ to s_i or t_i correspondingly, and when the robber moves back the cop also returns. Since $\phi = true$, we have that the cops in the $G_i(\exists)$ gadgets can move in such a way that when the robber occupies the vertex u_n, every vertex C_j has at least one neighbor that is occupied by a cop. If the robber moves to some vertex w_j then a cop moves to C_j, and if the robber moves back then this cop also moves back. Thus the cops have a winning strategy in this case. □

4.5 Proof of Lemma 3

Proof. Let us note that if the robber chooses u_0 as a starting point, then after this $2n+1$ cops have to occupy the vertices $p_0, p_1, \ldots, p_{2n}$. Also at least one cop has to protect the graph from a possible intrusion that can occur if the robber decides to start in b. Hence at the start of the game this cop is placed either on c_1, or on c_2 and can move only to vertices c_1, c_2, c_3 in his fist move. Notice also that if the robber moves from u_0 to u_n along some path then all these $2n+2$ cops cannot leave the set of vertices $\{c_1, c_2, c_3, p_0, q_0, p_1, q_1, \dots p_{2n}\}$ before the robber leaves u_n. This follows from the observation that if the cop from a vertex $x = p_{2i-1}$ leaves this vertex before the robber leaves y_i or $\overline{y}_i$ or the cop from the vertex $x = p_{2i}$ leaves this vertex before the robber leaves u_i, then the robber can enter x, because the cops which were standing in vertices $c_j, p_0, \dots, p_{i-1}$ in the beginning of the game can not "keep up" with the robber and reach the vertex x at this moment. Thus the $2n + 2$ cops that were added in the construction of G'' from G' are unable to block the vertices $C_1, C_2, \dots, C_m$. Also notice that the n cops initially placed on the vertices $s_1, \dots, s_n$ must behave exactly as they did in G'. Hence, Lemma 1 implies that these n cops can not guard the graph against the robber moving from u_0 in the direction of u_n.

Suppose now that $r \neq u_0$. We describe a winning strategy for the cops. In the beginning n cops occupy the vertices $s_1, s_2, \dots, s_n$, $2n + 1$ cops occupy vertices $q_0, q_1, \dots, q_{2n}$, and one cop is in c_2. If $r = b$ then the cop from c_2 moves to c_1 and the cop-player wins. If $r \neq b$ then the cop from c_2 moves to c_3, and by his next move he moves to p_0. The cops from $q_0, q_1, \dots, q_{2n-1}$ move to $p_1, p_2, \dots, p_{2n}$. The cop which occupies the vertex q_{2n} remains in it if the robber is on vertices u_i, y_i or $\overline{y}_i$. But if the robber moves (or chooses as a starting point) some vertex $a_i, \overline{a}_i$ or w_j or some vertex from W, then he moves to an adjacent vertex and prevents the robber from entering C''. If the robber moves back to vertices u_i, y_i or $\overline{y}_i$ then the cop returns to q_{2n}.

If $r = u_0$ and the robber can only move along the edges $u_{i-1}y_i, y_iu_i, u_{i-1}\overline{y}_i, \overline{y}_iu_i$ from the first vertex to the next, then the cops have a strategy which is winning when $\phi = true$. Cops start by occupying vertices $s_1, s_2, \dots, s_n$, This requires n cops, $2n + 1$ cops occupy $q_0, q_1, \dots, q_{2n}$, and one cop is placed on c_2. Notice that the cops have the same starting position as above. Now the cops from from $q_0, q_1, \dots, q_{2n}$ move to $p_0, p_1, \dots, p_{2n}$, and the cops from $s_1, s_2, \dots, s_n$ use the same strategy as in Lemma 2. □

Proof of Lemma 4

Proof. If $\phi = false$, then the robber can win by making use of exactly the same strategy as in Lemma 3. In this case, $3n + 2$ cops have to occupy the same vertices as in Lemma 3, namely the same vertices as before on $[G''; C'', R'']$ in the beginning of the game, and one cop has to occupy either e or r_0. Otherwise the robber can choose d and move to $e \in C$ in his first move. Note, that this cop cannot leave the vertices on the path P while the robber is moving from u_0 to u_n, since the robber can enter f_1 or f_2 otherwise. Notice also that if the robber moves from u_0 to u_n along some path in $G[C]$ then this cop cannot enter r_{2n+1} the moment the robber occupies u_n. Thus he cannot protect C from the robber.

Suppose now that $\phi = true$. We construct a winning strategy for the cop-player. At the beginning of the game, n cops occupy the vertices $s_1, s_2, \dots, s_n$, $2n + 1$ cops occupy vertices $q_0, q_1, \dots, q_{2n}$, and one cop is in c_2. The strategy for the cops is similar to the strategies in Lemmata 2 and 3. We place one cop on r_0. If the robber chooses d as his starting point, this cop moves to e and the cop-player wins. If the robber occupies the vertices u_i, y_i or $\overline{y}_i$, then the robber moves along P toward r_{2n+1}. If the robber moves to a vertex in W_1 or W_2, then the cop responds by moving to f_2 or f_1 respectively. Suppose that the robber made at least one "backward" move along edges $u_{i-1}y_i, y_iu_i, u_{i-1}\overline{y}_i, \overline{y}_iu_i$. If he tries to enter C by moving to some vertex a_i or $\overline{a}_i$, then the cop on the path P moves to z_i or $\overline{z}_i$ and then when the robber returns to y_i or $\overline{y}_i$ the cop moves to r_{2n+1}. In any case this cop reaches the vertex r_{2n+1} before the robber enters u_n, and from this vertex the cop can "block" all vertices $z_i, \overline{z}_i$ and every vertex C_j. □

4.6 Proof of Proposition 5

Proof. We have to prove that $\gamma(G[C], \delta[G; C, R], C\backslash\delta[G; C, R]) \geq \mathbf{gn}_1(G; C, R)$. Let X be a dominating set in the black and white graph $G[C]$ for $B = \delta[G; C, R]$). We define a strategy $\mathcal{S} = (s, \mathcal{F})$ for $c = |X|$ cops as follows. Let

$$s(v) = \begin{cases} 1, & \text{if } v \in X, \\ 0, & \text{if } v \notin X, \end{cases}$$

and for each $v \in B$, let $d(v)$ be an arbitrary vertex in $N_{G[C]}[v] \cap X$. For each vertex $u \in R$, we set $f_u(v) = d(v)$ if $v \in N_G(u) \cap C$. Since $g(G) \geq 5$, for any two different vertices $v, w \in N_G(u) \cap C$, $f_u(v) \neq f_u(w)$. Hence $\mathcal{S}$ is a winning strategy for c cops. □

Definition of treewidth

A *tree decomposition* of a graph G is a pair (T, X) where T is a tree whose vertices we will call *nodes* and $X = \{X_i \colon i \in V(T)\}$ is a collection of subsets of $V(G)$ (called *bags*) such that

1. $\bigcup_{i \in V(T)} X_i = V(G)$,
2. for each edge $xy \in E(G)$, there is an $i \in V(T)$ such that $x, y \in X_i$,
3. for each $x \in V(G)$ the set of nodes $\{i \colon x \in X_i\}$ forms a subtree of T.

The *width* of a tree decomposition equals $\max\{|X_i| - 1 \colon i \in V(T)\}$. The *treewidth* of a graph G (denoted by $\mathbf{tw}(G)$) is the minimum width over all tree decompositions of G.

Every tree decomposition can be easily converted (in linear time) to a *nice* tree decomposition of same width (and with a linear size of T) with the rooted binary tree T with the root r, which induces a parent-child relation in the tree, such that nodes of T are of four types:

1. *Leaf* nodes i are leaves of T and have $|X_i| = 1$.
2. *Introduce* nodes i have one child j with $X_i = X_j \cup \{v\}$ for some vertex $v \in V(G)$.
3. *Forget* nodes i have one child j with $X_i = X_j \setminus \{v\}$ for some vertex $v \in V(G)$.
4. *Join* nodes i have two children j_1 and j_2 with $X_i = X_{j_1} = X_{j_2}$.

Proof of Theorem 4

Proof. For any node $i \in V(T)$, we denote by T_i the rooted subtree induced by the descendants of i with the root i. We also define $G_i = G[\bigcup_{j \in V(T_i)} X_j]$, and set $Y_i = V(G_i) \setminus X_i$, $Z_i = V(G) \setminus V(G_i)$ and $C_i = C \cap V(G_i)$. Our algorithm constructs tables of data for trees T_i starting from the leaves. Since this approach is more or less standard dynamic programming on graphs of bounded treewidth (see e.g. survey [10]), we describe here only tables which are stored for nodes of T.

Let $\mathcal{U} = \{U_u\}_{u \in R}$ be a collection of sets (some sets can be equal) such that $U_u \subseteq N_G(u) \cap C \cap X_i$ for $u \in R$. The *partial strategy for cops* on G_i is defined as a pair $\mathcal{S}_i(\mathcal{U}) = (s, \mathcal{F})$ where

- s is a mapping assigning to every vertex v of C_i a non-negative integer $s(v)$,
- $\mathcal{F} = \{f_u\}_{u \in R}$ is a family of functions $f_u \colon C_i \cap N_G(u) \setminus U_u \to C_i$, such that for every $w \in C_i \cap N_G(u) \setminus U_u$, $f_u(w) \in N_G[w]$, and for every $v \in C_i$, $|\{w \in C_i \cap N_G(u) \colon f_u(w) = v\}| \leq s(v)$.

We call $s(C_i)$ the *weight* of partial strategy. Clearly, for the root r and the collection of empty sets $\mathcal{U}$, $\mathcal{S}_r(\mathcal{U})$ is a strategy for $s(C)$ cops.

For each partial strategy $\mathcal{S}_i(\mathcal{U}) = (s, \mathcal{F})$ of cops, $\mathcal{U} = \{U_u\}_{u \in R}$, and for a vertex $u \in R$, we define the *configuration of $\mathcal{S}_i(\mathcal{U})$ for u in X_i* as a 4-tuple $\{D_u, U_u, X_u, f'_u\}$ where sets $D_u, U_u, X_u \subseteq C \cap N_G(u) \cap X_i$ form a partition of $C \cap N_G(u) \cap X_i$ (some sets can be empty) such that $f_u(x) \in Y_i$ for $x \in D_u$, $f_u(x) \in X_i$ for $x \in X_u$, and $f'_u = f_u|_{X_u}$ (i.e. f'_u is the function on X_u such that $f'_u(x) = f_u(x)$ for $x \in X_u$).

Let $g_u(v) = s(v) - |\{x \in V(G_i) \colon f_u(x) = v\}|$ for $v \in C \cap X_i$ and $u \in R \cap X_i$. We define a function $s' = s|_{C \cap X_i}$.

The *configuration of $\mathcal{S}_i(\mathcal{U})$ for Y_i in X_i* is the set K_D of all different configurations of $\mathcal{S}_i(\mathcal{U})$ for $u \in R \cap Y_i$. Symmetrically, the *configuration of $\mathcal{S}_i(\mathcal{U})$ for Z_i in X_i* is the set K_U of all different configurations of $\mathcal{S}_i(\mathcal{U})$ for $u \in R \cap Z_i$. The *configuration of $\mathcal{S}_i(\mathcal{U})$ for X_i in X_i* is defined as the set K_X of all 6-tuples

$\{u, D_u, U_u, X_u, f'_u, g_u\}$ for $u \in R \cap X_i$. The *state* of the partial strategy $\mathcal{S}_i(\mathcal{U})$ for X_i is the 4-tuple $\{s', K_D, K_U, K_X\}$.

Correspondingly, the table of data for a node i of T_i contains all 5-tuples $\{w, s', K_D, K_U, K_X\}$, where $w \le n$ is a positive integer, $s' \colon C \cap X_i \to \{0, \dots, n\}$, K_D and K_U are sets of 4-tuples $\{D, U, X, f\}$ and K_X is a set of 6-tuples $\{\{u, D_u, U_u, X_u, f'_u, g_u\} \colon u \in R \cap X_i$. For each 4-tuple $\{D, U, X, f\}$ in K_D, $D, U, X \subseteq C \cap X_i$ which form a partition of the set $N_G(u) \cap X_i \cap C$ for some $u \in R \cap Y_i$, and for each $u \in R \cap Y_i$, K_D contains at least one 4-tuple such that D, U, X is a partition of $N_G(u) \cap X_i \cap C$. Respectively, for each 4-tuple $\{D, U, X, F\}$ in K_U, $D, U, X \subseteq C \cap X_i$ and they form a partition of the set $N_G(u) \cap X_i \cap C$ for some $u \in R \cap Z_i$, and for each $u \in R \cap Z_i$, K_D contains at least one 4-tuple such that D, U, X is a partition of $N_G(u) \cap X_i \cap C$. In both cases $f \colon X \to C \cap X_i$ such that $f(x) \in N_G[x]$ for $x \in X$. For each 6-tuple $\{u, D_u, U_u, X_u, f'_u, g_u\}$ in K_X, D_u, U_u, X_u is a partition of $N_G(u) \cap X_i \cap C$, $f'_u \colon X_u \to C \cap X_i$ such that $f'_u(x) \in N_G[x]$ for $x \in X_u$, and $g_u \colon X_i \cap C \to \{0, \dots, n\}$.

For each 5-tuple $\{w, s', K_D, K_U, K_X\}$, the table for the node i keeps "YES", if there is a partial strategy for G_i of weight w for some collection of sets $\mathcal{U}$ with this state, and the table contains "NO" otherwise.

Such tables can be constructed for leaves of T by trying all possible partial strategies, and it can be easily checked that the table for a vertex i can be computed if the tables for children of i are given. If the table for the root r is constructed, then we can find the value of $\mathbf{gn}_1(G; C, R)$.

A correctness of the algorithm follows from the description. Let us estimate the time complexity. The running time is proportional to the sizes of tables. Notice that the number of all possible 4-tuples in K_D (K_U respectively) is at most $p(t) = 2^{t+1} \cdot 3^{t+1} \cdot (t+1)^{t+1}$. Therefore, the number of all possible sets K_D (K_U respectively) is at most $2^{p(t)}$. The number of all possible 6-tuples in K_X for each $u \in R \cap X_i$ is $3^{t+1} \cdot (t+1)^{t+1} \cdot (n+1)^{t+1}$ at most, and hence there is at most $3^{(t+1)^2} \cdot (t+1)^{(t+1)^2} \cdot (n+1)^{(t+1)^2} = q(t) \cdot n^{(t+1)^2}$ possibilities to construct K_X. Hence, the number of 5-tuples $\{w, s', K_D, K_U, K_X\}$ in the table is $(n+1) \cdot (n+1)^{t+1} \cdot p(t)^2 \cdot q(t) \cdot n^{(t+1)^2}$ and the size of the table is bounded by the function $h(t) \cdot (n+1)^{t^2+3t+3}$. Thus the running time of the algorithm is bounded by $h(t) n^{O(t^2)}$ for some function h which depends only on t. □

Proof of Lemma 6

Proof. Suppose that $X \subseteq V(G)$ is a capacitated dominating set of size at most k. We assume without loss of generality that $|X| = k$ and $X = \{a_1, a_2, \dots, a_k\}$. We define a winning strategy for r cops $\mathcal{S} = (s, \mathcal{F})$ as follows. The function s on the vertices of $C(G_i)$ is defined as s_1 if $a_i \in X$ and s_2 if $a_i \notin X$ (see Lemma 5). For all other vertices t of C, we put $s(t) = 0$. Clearly, $s(C) = r$. Now we define $\mathcal{F}$. By Lemma 5, we have to define mappings $f_x \colon C \cap N_H(x) \to C$ only for $x = q_1, q_2, p$. Let $f_{q_1}(b_i) = a_i$ and $f_{q_2}(d_i) = a_i$ for all $i \in \{1, 2, \dots, k\}$. Denote by $g \colon \{a_{k+1}, a_{k+2}, \dots, a_n\} \to \{a_1, a_2, \dots, a_k\}$ a domination mapping of vertices of $V(G) \setminus X$ for c and X. We set $f_p(u_i) = u_i$ for $i \in \{1, 2, \dots, k\}$, and if $g(a_i) = a_j$ then $f_p(u_i) = v_j$ for $i > k$.

Assume now that $\mathbf{gn}_1(H; C, R) \le r$. Let $\mathcal{S} = (s, \mathcal{F})$ be a winning strategy for r cops. By the first claim of Lemma 5 $s(\bigcup\limits_{i=1}^{n} (C(G_i) \setminus \{u_i\}) \ge \sum\limits_{i=1}^{n} c(a_i))$, and then $s(\{u_1, u_2, \dots, u_n\} \cup \{b_1, b_2, \dots, b_k\} \cup \{d_1, d_2, \dots, d_k\}) \le k$. It can be easily seen that $f_{q_1}(b_i) \in \{b_i, u_1, u_2, \dots, u_u\}$ and hence $s(\{b_1, b_2, \dots, b_k\} \cup \{u_1, u_2, \dots, u_n\}) \ge k$. Similarly, $s(\{d_1, d_2, \dots, d_k\} \cup \{u_1, u_2, \dots, u_n\}) \ge k$. It follows that $s(\{u_1, u_2, \dots u_k\}) = k$. Let $X = \{a_i \colon s(u_i) \ge 1\}$. Clearly, $|X| \le k$. We prove that X is a capacitated dominating set. Since $s(\{u_1, u_2, \dots u_k\}) = k$, $s(\bigcup\limits_{i=1}^{n} (C(G_i) \setminus \{u_i\}) \le \sum\limits_{i=1}^{n} c(a_i))$. Therefore, by Lemma 5, $s(C(G_i) \setminus \{u_i\}) = c(a_i)$ for all $i \in \{1, 2, \dots, a_n\}$.Thus if $s(u_i) \ge 1$, then $s(v_i) \le c(a_i)$. By the second claim of Lemma 5, if $s(u_i) = 0$, then $s(v_i) = 0$. It follows immediately that if $s(u_i) = 0$, then $f_p(u_i) \in \{v_j \colon s(u_j) \ge 1\}$. We define the domination mapping g for X as follows: if $f_p(u_i) = v_j$ then $g(a_i) = a_j$ for $a_i \notin X$. □

Proof of Lemma 7

Proof. Let us look on the construction of H in a slightly different way. We can assume that we first construct a bipartite graph with vertices $v_1, v_2, \ldots, v_n$ and $u_1, u_2, \ldots, u_n$ such that vertices u_i and v_j are adjacent if and only if $a_i a_j \in E(G)$. The treewidth of this graph is at most $2 \cdot \mathbf{tw}(H) + 1$ because we can construct its tree decomposition replacing every vertex a_i in the bags of the tree decomposition of G by vertices u_i and v_i. Then vertices $b_1, b_2, \ldots, b_k$, $d_1, d_2, \ldots, d_k$ and p, q_1, q_2 are added, which increases treewidth by at most $2k+3$. The obtained graph has treewidth at most $2 \cdot \mathbf{tw}(H) + 1 + 2k + 3$. Clearly, the gluing of gadgets G_i to the pairs u_i, v_i does not increase the treewidth of H above this value. □

Proof of Theorem 6

Proof. The idea of the proof is to show that when the maximum vertex degree of G is bounded, the ONE-STEP GUARDING problem can be stated as an optimization problem which belongs to the LinEMSOL class (we refer to the paper of Arnborg et al. [6] for the definition of this class). As it was shown in [6], every problem expressible as an LinEMSOL problem is solvable in linear time on graphs of bounded treewidth. Or in other words, is fixed parameter tractable with linear dependence on the input length, when parameterized by the treewidth.

Because $\Delta(G) \leq d$, we can assume that $s(v) \leq d+1$ for $v \in C$ and for every strategy $\mathcal{S} = (s, \mathcal{F})$ of cops. It is convenient here to treat G as a directed graph with each undirected edge xy replaced by two directed edges (x, y) and (y, x). Denote by $A(G)$ the set of directed edges of G. The problem of computing the one-step guard number is the following minimization problem: Compute $\min |X_1| + 2|X_2| + \ldots + (d+1)|X_{d+1}|$, where $X_1, \ldots, X_{d+1}$ are pairwise disjoint subsets of C (X_i is a set of vertices such that each vertex is occupied by i cops). The sets $X_1, \ldots, X_{d+1}$ satisfy the following conditions: $\forall u \in R$, $\exists Y \in (X_1 \cup \ldots \cup X_{d+1}) \cap N_G(u)$ (Y is a set of vertices, where at least one cop remains) and $\exists R \subseteq A(G)$ (R is a set of directed edges corresponding to movements of the cops) such that $\forall v \in N_G(u) \cap C \setminus Y$, $\exists (w, v) \in R$ for which $w \in X_1 \cup \ldots \cup X_{d+1}$ and for each $w \in X_i \setminus Y$, $|\{(v, w) \colon (v, w) \in A_i\}| \leq i$ and for each $w \in X_i \cap Y$, $|\{(v, w) \colon (v, w) \in A_i\}| \leq i - 1$ for $i \in \{1, \ldots, d+1\}$. This yields that computing of the one-step guard number is expressible as an LinEMSOL problem. □

Proof of Lemma 8

Proof. Let $\mathcal{S}_1 = (s_1, \mathcal{F}_1)$ and $\mathcal{S}_2 = (s_2, \mathcal{F}_2)$ be strategies for $c_1 = \mathbf{gn}_1(G_1, C_1, R_1)$ cops on $[G_1;\, C_1, R_2]$ and for $c_2 = \mathbf{gn}_1(G_2, C_2, R_2)$ cops on $[G_2;\, C_2, R_2]$ correspondingly, $\mathcal{F}_1 = \{f_u^{(1)}\}$ and $\mathcal{F}_2 = \{f_u^{(2)}\}$. We define the strategy $\mathcal{S} = (s, \mathcal{F})$, $\mathcal{F} = \{f_u\}$ on $[G;\, C, R]$ as follows:

$$s(v) = \begin{cases} s_1(v), & \text{if } v \in C_1 \setminus C_2, \\ s_2(v), & \text{if } v \in C_2 \setminus C_1, \\ s_1(v) + s_2(v), & \text{if } v \in C_1 \cap C_2; \end{cases}$$

$f_u = f_u^{(1)}$ if $u \in R_1 \setminus R_2$, $f_u = f_u^{(2)}$ if $u \in R_2 \setminus R_1$, and for $u \in R_1 \cap R_2$,

$$f_u(v) = \begin{cases} f_u^{(1)}(v), \text{if } v \in C_1 \cap N_G(u), \\ f_u^{(2)}(v), \text{if } v \in (C_2 \setminus C_1) \cap N_G(u). \end{cases}$$

It is easy to check that $\mathcal{S}$ is a winning strategy for $c_1 + c_2$ cops. □

Proof of Theorem 8

Proof. For planar graphs it is assumed that plane embeddings are given, and we do not distinguish between a planar graph and it's plane embedding. Let G be a planar graph. The *layer decomposition* of G [7] is a disjoint partition of the vertex set $V(G)$ into sets $L_1, L_2, \ldots, L_r$, which are recursively defined as follows:

1. L_1 is the set of vertices on the exterior face of G;

2. L_i is the set of vertices on the exterior face of $G[V(G) \setminus \bigcup_{j=1}^{i-1} L_j]$ for $i \in \{2, 3, \ldots, r\}$.

The number r is called the *outerplanarity* of G. The layer decomposition of a planar graph (for given plane embedding) can be constructed efficiently.

Proposition 6 ([2]). *Let G be a planar graph with n vertices. The layer decomposition of G can be constructed it time $O(n)$.*

We also need the following property of the outerplanarity [11] (see also [2]).

Proposition 7 ([11]). *Let G be a planar graph with outerplanarity r and with n vertices. Then $\mathbf{tw}(G) \leq 3r - 1$ and tree decomposition of width at most $3r - 1$ can be found in time $O(rn)$.*

We are now in position to prove the theorem. Let $G' = G[N_G[\delta[G;\, C, R]]]$, $C' = C \cap N_G[\delta[G;\, C, R]]$ and $R' = R \cap N_G[\delta[G;\, C, R]]$. By Proposition 3 $\mathbf{gn}_1(G; C, R) = \mathbf{gn}_1(G'; C', R')$. Clearly, G' has outerplanarity at most 3. Then by Proposition 7, $\mathbf{tw}(G') \leq 8$ and by Theorem 4, $\mathbf{gn}_1(G'; C', R')$ can be computed polynomially. $\square$