Parameterized Analysis of the Cops and Robber Game

Harmender Gahlawat
Ben-Gurion University of the Negev, Beersheba, Israel

Meirav Zehavi
Ben-Gurion University of the Negev, Beersheba, Israel

Abstract

Pursuit-evasion games have been intensively studied for several decades due to their numerous applications in artificial intelligence, robot motion planning, database theory, distributed computing, and algorithmic theory. Cops and Robber (CnR) is one of the most well-known pursuit-evasion games played on graphs, where multiple *cops* pursue a single *robber*. The aim is to compute the *cop number* of a graph, k, which is the minimum number of cops that ensures the *capture* of the robber.

From the viewpoint of parameterized complexity, CnR is W[2]-hard parameterized by k [Fomin et al., TCS, 2010]. Thus, we study structural parameters of the input graph. We begin with the *vertex cover number* (vcn). First, we establish that $k \leq \frac{\mathsf{vcn}}{3} + 1$. Second, we prove that CnR parameterized by vcn is FPT by designing an exponential kernel. We complement this result by showing that it is unlikely for CnR parameterized by vcn to admit a polynomial compression. We extend our exponential kernels to the parameters *cluster vertex deletion number* and *deletion to stars number*, and design a linear vertex kernel for *neighborhood diversity*. Additionally, we extend all of our results to several well-studied variations of CnR.

2012 ACM Subject Classification Theory of computation → Parameterized complexity and exact algorithms; Mathematics of computing → Graph algorithms

Keywords and phrases Cops and Robber, Kernelization, Graph Searching, Fixed parameter tractability

Digital Object Identifier 10.4230/LIPIcs.MFCS.2023.49

Related Version *Full Version*: `https://doi.org/10.48550/arXiv.2307.04594` [29]

Funding This research was supported by the European Research Council (ERC) grant titled PARAPATH.

1 Introduction

In *pursuit-evasion*, a set of agents, called *pursuers*, plan to catch one or multiple *evaders*. Classically, pursuit-evasion games were played on geometric setups, where pursuers and evaders move on the plane [35, 49]. Parsons [48] formulated pursuit-evasion on graphs to model the search for a person trapped in caves, giving rise to the field of graph searching. Since then, pursuit-evasion has been studied extensively, having applications in artificial intelligence [36], robot motion planning [16, 40], constraint satisfaction and database theory [31, 32, 33], distributed computing [4, 18] and network decontamination [45], and significant implications in graph theory and algorithms [1, 25, 30, 54].

Cops and Robber (CnR) is one of the most intensively studied pursuit-evasion games on graphs, where a set of cops pursue a single robber. Players move in discrete time steps alternately, starting with the cops. In each move, a player can move to an adjacent vertex, and the cops win by *capturing* the robber (i.e., if a cop and the robber occupy the same vertex). The goal is to compute the *cop number* of a graph G, denoted $\mathsf{c}(G)$, which is the minimum number of cops required to win in G. We define the game formally in Section 2. CnR is well studied in the artificial intelligence literature under the name Moving Target

PURSUIT (MTP) [37], where we consider sub-optimal but faster strategies from an applicative point of view. The results have found numerous applications in game design, police chasing, path planning, and robot motion planning [5, 44, 56].

Determining the parameterized complexity of games is a well-studied research topic [10, 11, 52]. Most pursuit-evasion games are AW[*]-hard [53]. In particular, CNR is W[2]-hard parameterized by $\mathsf{c}(G)$ [27]. Thus, we consider structural parameterizations, focusing on *kernelization*, polynomial-time preprocessing with a parametric guarantee. Due to its profound impact, kernelization was termed "the lost continent of polynomial time" [26]. We begin with the most studied structural parameter: the *vertex cover number* (vcn) of the input graph. We bound $\mathsf{c}(G)$ in terms of vcn, as well as achieve both positive and negative results concerning the kernelization complexity of CNR parameterized by vcn. We generalize our kernelization results to the smaller parameters *cluster vertex deletion number* (cvd) and *deletion to stars number* (dts), as well as to the parameter *neighborhood diversity* (nd). Furthermore, we extend all our results to several well-studied variants of CNR.

The choice of vcn to study pursuit-evasion games is natural due to various scenarios where vcn is significantly smaller than the graph size. For example, this includes scenarios where we model the existence of one or few (possibly interconnected) central hubs – for illustration, suppose an intruder is hiding in a system of buildings where we have only few corridors but a large number of rooms, or suppose we have few virtual servers with many stations (e.g., of private users) that can communicate only with the servers. Furthermore, vcn is one of the most efficiently computable parameters from both approximation [55] and parameterized [17] points of view, making it fit from an applicative perspective even when a vertex cover is not given along with the input. Moreover, vcn is the best choice for proving negative results – indeed, our negative result on the kernelization complexity of CNR for vcn implies the same for many other well-known smaller parameters such as treewidth, treedepth and feedback vertex set [28]. One shortcoming of vcn as a parameter is that it is large for some simple (and easy to resolve) dense graphs like cliques. However, we generalize our kernel to cvd, which is small for these dense graphs, and to dts. Furthermore, we design a *linear* kernel for the well-studied parameter nd. We further discuss the utility of our kernels in the Conclusion.

Brief Survey. CNR was independently introduced by Quilliot [51] and Nowakowski and Winkler [46] with exactly one cop.[1] Aigner and Fromme [2] generalized CNR to multiple cops and defined the *cop number* of a graph. We refer to the book [9] for details.

The computational complexity of CNR is a challenging research subject. On the positive side, Berarducci and Intrigila [6] gave a backtracking algorithm to decide if G is k-copwin in $\mathcal{O}(n^{2k+1})$ time. On the negative side, Fomin et al. [27] proved that CNR is NP-hard, and W[2]-hard parameterized by k. Moreover, Mamino [43] showed that CNR is PSPACE-hard, and later, Kinnersley [39] proved that CNR is, in fact, EXPTIME-complete. Recently, Brandt et al. [14] proved that, conditioned on *(Strong) Exponential Time Hypothesis*, the time complexity of any algorithm for CNR is $(\Omega(n^{k-o(1)}))$ $2^{\Omega(\sqrt{n})}$. Since CNR admits an XP-time algorithm, it is sensible to bound the cop number for various graph classes or by structural parameters. Nowadays, we know that the cop number is at most 3 for toroidal graphs [41], 9 for unit-disk graphs [7], 13 for string graphs [20], and bounded for bounded genus graphs [12] and minor-free graphs [3]. Moreover, $\mathsf{c}(G) \leq \frac{\mathsf{tw}(G)}{2} + 1$ [38] and $\mathsf{c}(G) \leq \mathsf{cw}(G)$ [27], where $\mathsf{tw}(G)$ and $\mathsf{cw}(G)$ are the treewidth and the cliquewidth of G, respectively.

[1] In fact, a specific instance of CNR was given as a puzzle in the book [24] already in 1917.

Our Contribution. We conduct a comprehensive analysis of CnR parameterized by vcn. Due to space constraints, the proofs of the claims marked by (∗) and the claims for which we only provide a proof sketch are deferred to the full version [29].

We start by bounding the cop number of a graph:

▶ **Theorem 1.** *For a graph G, $\mathsf{c}(G) \leq \frac{\mathsf{vcn}}{3} + 1$.*

The proof is based on the application of three reduction rules. Each of our rules controls its own cop, which guards at least three vertices from the vertex cover. Once our rules are no longer applicable, we exhibit that the remaining unguarded part of the graph is of a special form, and the usage of only two additional cops suffices. We complement Theorem 1 with an argument that it might be difficult to improve this bound further using techniques similar to ours.

Second, we prove that CnR parameterized by vcn is FPT by designing a kernelization algorithm:

▶ **Theorem 2.** *CnR parameterized by* vcn *admits a kernel with at most* $\mathsf{vcn} + \frac{2^{\mathsf{vcn}}}{\sqrt{\mathsf{vcn}}}$ *vertices.*

Our kernel is also based on the application of reduction rules. However, these rules are very different than those used for the proof of Theorem 1. While our main rule is quite standard in kernelization (involving the removal of false twins), the proof of its correctness is (arguably) not. Theorem 2, Theorem 1, and the XP-algorithm (Proposition 12) yield the following corollary:

▶ **Corollary 3.** *CnR is* FPT *parameterized by* vcn*, and solvable in* $\left(\mathsf{vcn} + \frac{2^{\mathsf{vcn}}}{\sqrt{\mathsf{vcn}}}\right)^{\frac{\mathsf{vcn}}{3}+2} \cdot n^{\mathcal{O}(1)}$ *time.*

We complement our kernel by showing that it is unlikely for CnR to admit polynomial compression, by providing a *polynomial parameter transformation* from Red-Blue Dominating Set. Our reduction makes non-trivial use of a known construction of a special graph having high girth and high minimum degree.

▶ **Theorem 4.** *CnR parameterized by* vcn *does not admit polynomial compression, unless* NP ⊆ coNP/poly.

We also present a linear vertex kernel parameterized by nd for CnR, Lazy CnR, and Cops and Attacking Robber, and a quadratic vertex kernel for Cops and Fast Robber and Fully Active CnR:

▶ **Theorem 5** (∗)**.** *CnR, Lazy CnR, and Cops and Attacking Robber parameterized by* nd *admits a kernel with at most* nd *vertices. Moreover, Cops and Fast Robber and Fully Active CnR parameterized by* nd *admit a kernel with at most* nd^2 *vertices.*

On the positive side, we extend our exponential kernel to smaller parameters, cvd and dts:

▶ **Theorem 6.** *CnR parameterized by* cvd *admits a kernel with at most* $2^{2^{\mathsf{cvd}}+\sqrt{\mathsf{cvd}}}$ *vertices. Moroever, CnR parameterized by* dts *admits a kernel with at most* $2^{2^{\mathsf{dts}}+\mathsf{dts}^{1.5}}$ *vertices.*

Several variants of CnR have been studied due to their copious applications. We extend our results, parameterized by vcn, to some of the most well-studied ones. We define these variants (and used notations) in Section 2. We first bound the cop number of these variants by vcn:

▶ **Theorem 7** ($*$)**.** *For a graph G: (1) $\mathsf{c}_{lazy} \leq \frac{\mathsf{vcn}}{2} + 1$; (2) $\mathsf{c}_{attack} \leq \frac{\mathsf{vcn}}{2} + 1$; (3) $\mathsf{c}_{active}(G) \leq \mathsf{vcn}$; (4) $\mathsf{c}_{surround}(G) \leq \mathsf{vcn}$; (5) $\mathsf{c}_s(G) \leq \mathsf{vcn}$ (for any value of s); (6) for a strongly connected orientation $\overrightarrow{G}$ of G, $\mathsf{c}(\overrightarrow{G}) \leq \mathsf{vcn}$.*

We also extend our exponential kernel to these variants:

▶ **Theorem 8.** *Cops and Attacking Robber and Lazy CnR parameterized by vcn admit a kernel with at most $\mathsf{vcn} + \frac{2^{\mathsf{vcn}}}{\sqrt{\mathsf{vcn}}}$ vertices. Moreover, CnR on strongly connected directed graphs admits a kernel with at most $3^{\mathsf{vcn}} + \mathsf{vcn}$ vertices.*

Then, we present a slightly more general kernelization that works for most variants of the game:

▶ **Theorem 9.** *Fully Active CnR, Cops and Fast Robber, and Surrounding CnR parameterized by vcn admit a kernel with at most $\mathsf{vcn} + \mathsf{vcn} \cdot 2^{\mathsf{vcn}}$ vertices.*

We complement our exponential kernels for these variants by arguing about their compressibility:

▶ **Theorem 10** ($*$)**.** *Lazy CnR, Cops and Attacking Robber, Cops and Fast Robber, Fully Active CnR, and CnR on strongly connected directed and oriented graphs parameterized by vcn do not admit a polynomial compression, unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$.*

2 Preliminaries

For $\ell \in \mathbb{N}$, let $[\ell] = \{1, \ldots, \ell\}$. Whenever we mention $\frac{a}{b}$, we mean $\lceil \frac{a}{b} \rceil$.

Graph Theory. We only consider finite, connected[2], and simple graphs. For a graph G, we denote $|V(G)|$ by n. Let $v \in V(G)$. Then, $N(v) = \{u \mid uv \in E(G)\}$ and $N[v] = N(v) \cup \{v\}$. For $X \subseteq V(G)$, let $N_X(v) = N(v) \cap X$ and $N_X[v] = N[v] \cap X$. We say that v *dominates* u if $u \in N[v]$. The *girth* of G is the length of a shortest cycle contained in G. A *u,v-path* is a path with endpoints u and v. A path is *isometric* if it is a shortest path between its endpoints. For a directed graph $\overrightarrow{G}$, let $N^+(u)$ and $N^-(u)$ denote the set of out-neighbors and in-neighbors of u, respectively.

Let G be a graph and $U \subseteq V(G)$. Then, $G[U]$ denotes the subgraph of G induced by U. A set $U \subseteq V(G)$ is a *vertex cover* if $G[V(G) \setminus U]$ is an independent set. The minimum cardinality of a vertex cover of G is its *vertex cover number* (vcn). Moreover, U is a *cluster vertex deletion set* if $G[V(G) \setminus U]$ is a disjoint union of cliques. The minimum size of a cluster vertex deletion set of a graph is its *cluster vertex deletion number* (cvd). Additionally, U is a *deletion to stars set* if $G[V(G) \setminus U]$ is a disjoint union of star graphs. The minimum size of a deletion to stars set of a graph is its *deletion to stars number* (dts). Two vertices $u, v \in V(G)$ have the *same type* if and only if $N(v) \setminus \{u\} = N(u) \setminus \{v\}$. A graph G has *neighborhood diversity* at most w if there exists a partition of $V(G)$ into at most w sets, such that all the vertices in each set have the same type.

[2] The cop number of a disconnected graph is the sum of the cop numbers of its components; hence, we assume connectedness.

CnR. CNR is a two-player perfect information pursuit-evasion game played on a graph. One player is referred as *cop player* and controls a set of *cops*, and the other player is referred as *robber player* and controls a single *robber*. The game starts with the cop player placing each cop on some vertex of the graph, and multiple cops may simultaneously occupy the same vertex. Then, the robber player places the robber on a vertex. Afterwards, the cop player and the robber player make alternate moves, starting with the cop player. In the cop player move, the cop player, for each cop, either moves it to an adjacent vertex (along an edge) or keeps it on the same vertex. In the robber player move, the robber player does the same for the robber. For simplicity, we will say that the cops (resp., robber) move in a cop (resp., robber) move instead of saying that the cop (resp., robber) player moves the cops (resp., robber). Throughout, we denote the robber by $\mathcal{R}$.

A situation where one of the cops, say, $\mathcal{C}$, occupies the same vertex as $\mathcal{R}$ is a *capture*. The cops win if they have a strategy to capture $\mathcal{R}$, and $\mathcal{R}$ wins if it has a strategy to evade a capture indefinitely. A graph G is k-*copwin* if k cops have a winning strategy in G. The *cop number* of G, denoted $\mathsf{c}(G)$, is the minimum k such that G is k-copwin. For brevity, G is said to be *copwin* if it is 1-copwin (i.e. $\mathsf{c}(G) = 1$). Given a graph G and $k \in \mathbb{N}$, CNR asks whether G is k-copwin.

If a cop $\mathcal{C}$ occupies a vertex v, then $\mathcal{C}$ *attacks* $N[v]$. A vertex u is *safe* if it is not being attacked by any cop. If $\mathcal{R}$ is on a vertex that is not safe, then $\mathcal{R}$ is *under attack*. We say that some cops *guard* a subgraph H of G if $\mathcal{R}$ cannot enter H without getting captured by one of these cops in the next cop move. We shall use the following result:

▶ **Proposition 11** ([2]). *One cop can guard an isometric path P after a finite number of cop moves.*

Currently, the best known algorithm to decide whether G is k-copwin is by Petr et al. [50]:

▶ **Proposition 12** ([50]). *CNR is solvable in $\mathcal{O}(kn^{k+2})$ time.*

Variations of CnR. Several variations of CNR have been studied in the literature, differing mainly in the rules of movements of agents, the definition of the capture, and the capabilities of the agents. We provide below the definitions of the games considered in this paper here. A detailed overview of these games is provided in the full version [29].

Lazy CnR. In LAZY CNR [47], the cops are *lazy*, i.e., at most one cop can move during the cops' turn. The cop number in this game is denoted by $\mathsf{c}_{lazy}(G)$.

Cops and Attacking Robber. In COPS AND ATTACKING ROBBER [8], if on a robber's turn, there is a cop in its neighborhood, then $\mathcal{R}$ can attack the cop and *eliminate* it from the game. However, if more than one cop occupy a vertex and $\mathcal{R}$ attacks them, then only one of the cops gets eliminated, and $\mathcal{R}$ gets captured by one of the remaining cops. Here, the cop number is denoted by $\mathsf{c}_{attack}(G)$.

Fully Active CnR. In FULLY ACTIVE CNR [34], in a cop/robber move, each cop/robber must move to an adjacent vertex. Here, the cop number is denoted by $\mathsf{c}_{active}(G)$.

Surrounding CnR. SURROUNDING CNR [15] differs in the definition of capture. Here, a cop and $\mathcal{R}$ can occupy the same vertex during the game, but $\mathcal{R}$ cannot end its turn by remaining at a vertex occupied by some cop. The cops win by *surrounding* $\mathcal{R}$, i.e., if $\mathcal{R}$ occupies a vertex v, then there is a cop at each vertex $u \in N(v)$. The *surrounding cop number* for G is denoted as $\mathsf{c}_{surround}(G)$.

Cops and Fast Robber. In Cops and Fast Robber [27], $\mathcal{R}$ can move faster than the cops. If $\mathcal{R}$ has speed s, then it can move along a path with at most s edges not containing any cop. The minimum number of cops that ensures capture of a fast robber with speed s in a graph G is denoted by $\mathsf{c}_s(G)$. For $s \geq 2$, deciding whether $\mathsf{c}_s(G) \leq k$ is W[2]-hard parameterized by k even when input graph G is restricted to be a split graph [27].

CnR on Directed Graphs. In the game of CnR on directed graphs [42, 13, 21], the players can only move along the orientation of the arcs.

An XP Algorithm for Variants. For graph searching, there is a standard technique to get an $n^{\mathcal{O}(k)}$ time XP algorithm. This technique involves generating a *game graph* where each vertex represents a possible placement of all the agents (game states) on the vertices of G. Petr et al. [50] implemented this algorithm for CnR in $\mathcal{O}(kn^{k+2})$ time. It is not difficult to see that this algorithm can be made to work for all the variants we discussed (by changing the rules to navigate between game states). We discuss these extensions in the full version [29].

▶ **Proposition 13.** *Any variant of CnR considered in this paper is solvable in $\mathcal{O}(kn^{k+2})$ time.*

Parameterized Complexity. In the framework of parameterized complexity, each problem instance is associated with a non-negative integer, called a *parameter*. A parametrized problem Π is *fixed-parameter tractable* (FPT) if there is an algorithm that, given an instance (I, k) of Π, solves it in time $f(k) \cdot |I|^{\mathcal{O}(1)}$ for some computable function $f(\cdot)$. Two instances I and I' are *equivalent* when I is a Yes-instance if and only if I' is a Yes-instance. A *compression* of a parameterized problem Π_1 into a (possibly non-parameterized) problem Π_2 is a polynomial-time algorithm that maps each instance (I, k) of Π_1 to an equivalent instance I' of Π_2 such that size of I' is bounded by $g(k)$ for some computable function $g(\cdot)$. If $g(\cdot)$ is polynomial, then the problem is said to admit a *polynomial compression*. A *kernelization algorithm* is a compression where $\Pi_1 = \Pi_2$. Here, the output instance is called a *kernel*. Let Π_1 and Π_2 be two parameterized problems. A *polynomial parameter transformation* from Π_1 to Π_2 is a polynomial-time algorithm that, given an instance (I, k) of Π_1, generates an equivalent instance (I', k') of Π_2 such that $k' \leq p(k)$, for some polynomial $p(\cdot)$. It is well-known that if Π_1 does not admit a polynomial compression, then Π_2 does not admit a polynomial compression [17]. We refer to the books [17, 28] for details on parameterized complexity.

3 Bounding the Cop Number

We will use the following lemma to derive upper bounds on $\mathsf{c}(G)$ for several graph parameters.

▶ **Lemma 14** ($*$)**.** *Let G be a graph and let $U \subseteq V(G)$ be a set of vertices such that for each connected component H of $G[V(G) \setminus U]$, $\mathsf{c}(H) \leq \ell$. Then, $\mathsf{c}(G) \leq \frac{|U|}{2} + \ell$.*

Since star graphs and complete graphs are copwin, Lemma 14 implies the following theorem.

▶ **Theorem 15.** *Let G be a graph and $t = \min\{\mathsf{cvd}, \mathsf{dts}\}$. Then, $\mathsf{c}(G) \leq \frac{t}{2} + 1$.*

Bounding Cop Number by vcn. Let U be a vertex cover of size t in G and I be the independent set $V(G) \setminus U$. Lemma 14 implies that $\mathsf{c}(G) \leq \lceil \frac{t}{2} \rceil + 1$. In this section, we improve this bound. First, we provide the following reduction rules.

▶ **Reduction Rule 1** (RR1). *If there is a vertex $v \in I$ such that $|N(v)| \geq 3$, then place a cop at v and delete $N[v]$.*

▶ **Reduction Rule 2** (RR2). *If there is a vertex $v \in U$ such that $|N[v] \cap U| \geq 3$, then place a cop at v and delete $N[v]$.*

▶ **Reduction Rule 3** (RR3). *If there is an isometric path P such that P contains at least three vertices from U, then guard P using one cop and delete $V(P)$ (see Proposition 11).*

We note the following.

▶ Note 16. In the application of RR1-RR3, whenever a set of vertices $X \subseteq V(G)$ is deleted by the application of RR1-RR3, it implies that each vertex $x \in X$ is being guarded by some cop, and hence, is not accessible to $\mathcal{R}$. We do not actually delete the vertices, and this deletion part is just for the sake of analysis. Hence, from the cop player's perspective, the graph remains connected.

Next, we argue that, after an exhaustive application of RR1-RR3 on G, each connected component has a special structure and is 2-copwin.

▶ **Lemma 17** ($*$). *Once we cannot apply RR1-RR3 anymore, let $\mathcal{R}$ be in a connected component H of G. Then, $c(H) \leq 2$.*

Finally, we have the following theorem.

▶ **Theorem 1.** *For a graph G, $\mathsf{c}(G) \leq \frac{\mathsf{vcn}}{3} + 1$.*

Proof Sketch. The proof follows from Lemma 17 and the fact that in each application of RR1-RR3, we use one cop to remove (or guard) at least three new vertices from U. ◀

We note that a similar technique will fail if we try to "remove" four vertices in each reduction rule [29].

4 Exponential Kernels

Exponential Kernel by vcn. Let G be a graph where a vertex cover U of size t is given. If no such vertex cover is given, then we can compute a vertex cover U of size $t \leq 2 \cdot \mathsf{vcn}$ using a polynomial-time approximation algorithm [55]. Let I be the independent set $V(G) \setminus U$. Our kernelization algorithm is based on the exhaustive application of the following reduction rules.

▶ **Reduction Rule 4** (RR4). *If $k > \frac{t}{3}$, then answer positively.*

▶ **Reduction Rule 5** (RR5). *If $k = 1$, then apply an $\mathcal{O}(n^3)$ time algorithm (Proposition 12) to check whether G is copwin.*

▶ **Reduction Rule 6** (RR6). *If there are two distinct vertices $u, v \in I$ such that $N(u) \subseteq N(v)$, then delete u.*

The safeness of RR4 follows from Theorem 1. The following lemma proves the safeness of RR6. We note that Lemma 18 can also be derived from [6, Corollary 3.3], but we give a self-contained proof for the sake of completeness.

▶ **Lemma 18.** *Let u and v be two distinct vertices of G such that $N(u) \subseteq N(v)$. Consider the subgraph H of G induced by $V(G) \setminus \{u\}$. Let $k \geq 2$. Then, G is k-copwin if and only if H is k-copwin.*

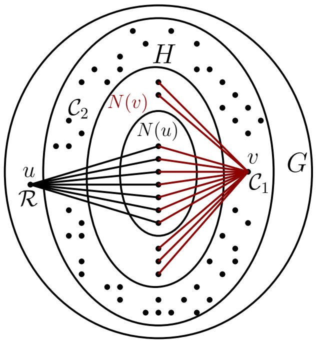

Figure 1 Illustration for Lemma 18. Here, $\mathcal{R}$ is at vertex u and $\mathcal{C}_1$ is at vertex v.

Proof. First, let G be k-copwin. Then, for the graph H, the k cops borrow the winning strategy that they have for G, with the only difference that whenever a cop has to move to the vertex u in G, it moves to v (in H) instead. Since $N(u) \subseteq N(v)$, the cop can make the next move as it does in the winning cop strategy for G. Note that using this strategy, the cops can capture $\mathcal{R}$ if $\mathcal{R}$ is restricted to $V(H)$ in G. Therefore, using this strategy, k cops will capture $\mathcal{R}$ in H as well.

Second, we show that if H is k-copwin, then G is is k-copwin. Here, for each vertex $x \neq u$ of G, we define $I(x) = x$, and for u, we define $I(u) = v$. Observe that for each $x \in V(G)$, $I(x)$ is restricted to H and if $xy \in E(G)$, then $I(x)I(y) \in E(H)$. Therefore, every valid move of a player from a vertex x to y in G can be translated to a valid move from $I(x)$ to $I(y)$ in H. Now, the cops have the following strategy. If the robber is on a vertex x, the cops consider the *image* of the robber on the vertex $I(x)$. Since the robber's image is restricted to H, the cops can use the winning strategy for H to capture the image of the robber in G. Once the image is captured, if the robber is not on the vertex u, then the robber is also captured. Otherwise, the robber is on the vertex u, and at least one cop is on v. See Figure 1 for an illustration. So, one cop, say $\mathcal{C}_1$, stays on v and this prevents the robber from ever leaving u. Indeed this follows because $N(u) \subseteq N(v)$, and so, if $\mathcal{R}$ ever leaves u, it will be captured by $\mathcal{C}_1$ in the next cop move. Finally, since $k > 1$, some other cop, say $\mathcal{C}_2$, can use a finite number of moves to reach u and capture the robber. ◀

Note that the requirement for $k \geq 2$ in Lemma 18 is crucial. Otherwise, we can get an H such that $c(H) = 1$, but $\mathsf{c}(G) > 1$. For example, consider C_4, where any two non-adjacent vertices satisfy the property in RR6, and if we remove one of them, the cop number reduces from 2 to 1. However, this does not harm our algorithm because if we are given $k = 1$, then RR5 is applied (before RR6).

Two sets A and B are *incomparable* if neither $A \subseteq B$ nor $B \subseteq A$. We shall use the following proposition that follows from Sperner's Theorem and Stirling's approximation.

▶ **Proposition 19.** *Let X be a set of cardinality N. Moreover, let Y be a set of subsets of X such that for each $a, b \in Y$, a and b are incomparable. Then, $|Y| \leq \frac{2^N}{\sqrt{N}}$.*

Once we cannot apply RR4-RR6 anymore, we claim that the size of the reduced graph G' is bounded by a function of t. Let $U' = U \cap V(G')$ and $I' = I \cap V(G')$. Clearly, $|U'| \leq t$. Now, each vertex $u \in I'$ is associated with a neighborhood $N(u)$ such that $N(u) \subseteq U'$. Moreover, for any two vertices $u, v \in I'$, the sets $N(u)$ and $N(v)$ are incomparable. Hence, due to Proposition 19, $|I'| \leq \frac{2^t}{\sqrt{t}}$, and therefore, $|V(G')| \leq t + \frac{2^t}{\sqrt{t}}$, which proves the following theorem.

▶ **Theorem 2.** *CnR parameterized by* vcn *admits a kernel with at most* $\mathsf{vcn} + \frac{2^{\mathsf{vcn}}}{\sqrt{\mathsf{vcn}}}$ *vertices.*

The details for exponential kernels for other parameters and variants can be found in [29]. Here, we present only our reduction rules and the essential claims that lead to our results.

Kernel by cvd. Let U be a cluster vertex deletion set of size t. Let $S = V(G) \setminus U$, and $C_1, \ldots, C_\ell$ be the set of disjoint cliques in $G[S]$. We have the following reduction rules along with RR 5.

▶ **Reduction Rule 7** (RR7). *If* $k > \frac{t}{2}$, *then answer positively.*

▶ **Reduction Rule 8** (RR8). *Let* u *and* v *be vertices of some clique* $C \in G[S]$ *such that* $N[u] \subseteq N[v]$. *Then, delete* u.

▶ **Reduction Rule 9** (RR9). *Let* C_i *and* C_j *be two cliques in* $G[S]$ *such that for each vertex* $u \in V(C_i)$, *there exists a vertex* $v \in V(C_j)$ *such that* $N_U(u) \subseteq N_U(v)$. *Then, delete* $V(C_i)$.

RR7 is safe due to Theorem 15. The safeness of RR8 is proved by the following Lemma, whose proof is similar to the proof of Lemma 18.

▶ **Lemma 20** (∗). *Let* u *and* v *be vertices of some clique* C *of* $G[S]$. *If* $N_U(u) \subseteq N_U(v)$, *then* $\mathsf{c}(G) = \mathsf{c}(G[V(G) \setminus \{u\}])$.

The safeness of RR9 is proved by the following Lemma.

▶ **Lemma 21** (∗). *Let* C_i *and* C_j *be two cliques in* $G[S]$ *such that for each vertex* $u \in V(C_i)$, *there exists a vertex* $v \in V(C_j)$ *such that* $N_U(u) \subseteq N_U(v)$. *Then, for* $k > 1$, G *is* k*-copwin if and only if* $G[V(G) \setminus V(C_i)]$ *is* k*-copwin.*

Finally, we use the following lemma to bound the size of the desired kernel from Theorem 6.

▶ **Lemma 22** (∗). *After an exhaustive application of RR7-RR9, the size of the reduced graph is at most* $2^{2^t+\sqrt{t}}$.

Kernel by dts. Using similar ideas, we can also get a kernel for CnR parameterized by dts. Let U be a deletion to stars vertex set of size t. Also, let $S = V(G) \setminus U$, and let $X_1, \ldots X_\ell$ be the stars in the graph $G[S]$. Specifically, we have the following reduction rules along with RR7.

▶ **Reduction Rule 10** (RR10). *Let* u *and* v *be two leaves of a star* X *in* $G[S]$ *such that* $N_U(u) \subseteq N_U(v)$. *Then, delete* u.

▶ **Reduction Rule 11** (RR11). *Let* X *and* Y *be two stars in* $G[S]$ *such that* $V(X) = x, x_1, \ldots, x_p$ *and* $V(Y) = y, y_1, \ldots, y_q$, *where* x *and* y *are center vertices of* X *and* Y, *respectively. If* $N_U(x) \subseteq N_U(y)$ *and for each vertex* x_i *(for* $i \in [p]$*), there is a vertex* y_j *(for* $j \in [q]$*) such that* $N_U(x_i) \subseteq N_U(y_j)$, *then delete* X.

The following lemma establishes that RR10 and RR11 are safe. (RR7 is safe due to Theorem 15.)

▶ **Lemma 23** (∗). *Assuming* $k > 1$, *RR10 and RR11 are safe.*

Using calculations similar to ones used in our previous kernels and Proposition 19, we can establish that (see [29]), once we cannot apply RR10 and RR11 anymore, the size of the reduced graph can be at most $\frac{2^{2^t}}{\sqrt{2^t}} \cdot 2^t \cdot (\frac{2^t}{\sqrt{t}} + 1)$, giving us the desired kernel from Theorem 6.

Kernels for Different Variants

Lazy CnR and Cops and Attacking Robber. For these variants, we have the following reduction rules along with RR6, which we apply before applying RR6.

▶ **Reduction Rule 12** (RR12). *If $k \geq \frac{t}{2} + 1$, then answer positively (Theorem 7).*

▶ **Reduction Rule 13** (RR13). *If $k = 1$, then apply the $\mathcal{O}(n^3)$ time algorithm from Proposition 13.*

The next lemma proves safeness of RR6 for both variants.

▶ **Lemma 24** (∗). *Let u and v be two distinct vertices of G such that $N(u) \subseteq N(v)$. Consider the graph H induced by $V(G) \setminus \{u\}$. Then for $k > 1$ and for $x \in \{lazy, attack\}$, $\mathsf{c}_x(G) \leq k$ if and only if $\mathsf{c}_x(H) \leq k$.*

The size of the kernel, by using these reduction rules, is dependent on RR6. Therefore, we get the desired kernels for these variants as claimed in Theorem 8.

CnR on Directed Graphs. Next, we consider the game of CnR on directed graphs. We have the following lemma.

▶ **Lemma 25** (∗). *Let u and v be two distinct vertices of a strongly connected directed graph $\overrightarrow{G}$ such that $N^+(u) \subseteq N^+(v)$ and $N^-(u) \subseteq N^-(v)$. Let $\overrightarrow{H}$ be the graph induced by $V(\overrightarrow{G}) \setminus \{u\}$. Then, for $k > 1$, $\overrightarrow{H}$ is k-copwin if and only if $\overrightarrow{G}$ is k-copwin.*

Let G be a graph with a vertex cover U of size t, and let $I = V(G) \setminus U$. Let $\overrightarrow{G}$ be a strongly connected orientation of G. We apply the following reduction rules.

▶ **Reduction Rule 14** (RR14). *If $k \geq t$, then answer positively.*

▶ **Reduction Rule 15** (RR15). *If $k = 1$, then apply the $\mathcal{O}(n^3)$ time algorithm (Proposition 13) to check if $\overrightarrow{G}$ is copwin.*

▶ **Reduction Rule 16** (RR16). *If u and v are two distinct vertices in I such that $N^+(u) \subseteq N^+(v)$ and $N^-(u) \subseteq N^-(v)$, then delete u.*

Safeness of RR14 and RR16 follow from Theorem 7 and Lemma 25, respectively. Once we cannot apply RR16, observe that each vertex $u \in I$ has a unique neighborhood $(N^+(u) \cup N^-(u))$, and there are three choices for a vertex $v \in U$ to appear in the neighborhood of u, i.e., either $v \in N^+(u)$, or $v \in N^-(u)$, or $v \notin N^+(u) \cup N^-(u)$. Thus, $|I| \leq 3^t$, giving the desired kernel as claimed in Theorem 8.

General Kernelization. Here, we provide a general reduction rule that works for most variants of CnR parameterized by vcn. Let U be a vertex cover of size t in G and I be the independent set $V(G) \setminus U$. For each subset $S \subseteq U$, we define the following equivalence class: $\mathcal{C}_S = \{v \in I \colon N(v) = S\}$. We have the following reduction rule along with RR14.

▶ **Reduction Rule 17** (RR17). *If there is an equivalence class $\mathcal{C}_S$ such that $|\mathcal{C}_S| > k + 1$, then keep only $k + 1$ arbitrary vertices from $\mathcal{C}_S$ in G, and delete the rest.*

The following lemma proves the safeness of RR17.

▶ **Lemma 26** (∗). *Let G be a graph with a vertex cover U of size t. Let $\mathcal{C}_S$ (for $S \subseteq U$) be an equivalence class such that $|\mathcal{C}_S| = \ell > k+1$. Moreover, let H be a subgraph formed by deleting $\ell - k - 1$ arbitrary vertices of $\mathcal{C}_S$ from G. Then,*

1. $\mathsf{c}_{active}(H) \leq k$ *if and only if* $\mathsf{c}_{active}(G) \leq k$.
2. $\mathsf{c}_s(H) \leq k$ *if and only if* $\mathsf{c}_s(G) \leq k$, *for any* $s \geq 1$.
3. $\mathsf{c}_{surround}(H) \leq k$ *if and only if* $\mathsf{c}_{surround}(G) \leq k$.

Proof Sketch. Here, we present a sketch of the proof for (3). Let $\mathcal{C}_S = \{v_1, \ldots, v_\ell\}$. WLOG, let the vertices $v_1, \ldots v_{k+1}$ belong to the graph H and vertices $v_{k+2}, \ldots, v_\ell$ are deleted. Note that $\mathcal{R}$ cannot be surrounded at a vertex in S in G since each vertex in S has at least $k+1$ neighbours.

The proof of $\mathsf{c}_{surround}(H) \leq \mathsf{c}_{surround}(G)$ is similar to the proof of Lemma 18.

For the other direction, let $\mathsf{c}_{surround}(H) \leq k$. Since we have only k cops, at any time, there is at least one vertex in $\{v_1, \ldots, v_{k+1}\}$ that is not occupied by any cop. Let us call this vertex a *free vertex* (there might be multiple free vertices). For each vertex $x \in V(G)$, if $x \in V(H)$, then we define $I(x) = x$; else, if $x \in \{v_{k+1}, \ldots v_\ell\}$, then we define $I(x) = y$, where y is a free vertex at that instance. Whenever $\mathcal{R}$ moves to a vertex $x \in V(G)$, we say that the *image of the robber*, denoted $\mathcal{I}_\mathcal{R}$, moves to $I(x)$. Recall that, in this variant, although some cop and $\mathcal{R}$ can be at the same vertex, $\mathcal{R}$ cannot end its move at the same vertex as one of the cops. Cops use this capability to force $\mathcal{R}$ to move from a vertex. Therefore, we also have to argue that whenever cops force $\mathcal{R}$ to move, they force $\mathcal{I}_\mathcal{R}$ to move as well. To this end, observe that $\mathcal{I}_\mathcal{R}$ and $\mathcal{R}$ are on different vertices only if $\mathcal{R}$ is on some vertex $x \in \{v_{k+1}, \ldots, v_\ell\}$ and $\mathcal{I}_\mathcal{R}$ is on a free vertex, say, y. Notice that if, in the strategy for H, $\mathcal{R}$ was occupying y and the cop player wants to force $\mathcal{R}$ to move out of y, then it does so by moving a cop, say, $\mathcal{C}$, from a vertex $w \in N(y)$ to y. Cop player adapts this strategy in G by moving $\mathcal{C}$ form w to x instead of w to y (it is possible because $N(x) = N(y)$). Thus, $\mathcal{R}$, as well as $\mathcal{I}_\mathcal{R}$, are forced to move as they would have been forced to move in the winning strategy of k cops in H.

Hence, $\mathcal{I}_\mathcal{R}$ is restricted to $V(H)$ in G. Thus, cops will surround $\mathcal{I}_\mathcal{R}$ in a finite number of rounds. The only thing to observe now is that if $\mathcal{I}_\mathcal{R}$ is surrounded in $V(H)$, then $\mathcal{R}$ is surrounded in G. ◀

Lemma 26 and the following lemma imply Theorem 9.

▶ **Lemma 27** (∗). *Let G be a graph with a vertex cover U of size t. After an exhaustive application of RR14 and RR17, the reduced graph has at most $t + t \cdot 2^t$ vertices.*

5 Incompressibility

In this section, we show that it is unlikely for CnR parameterized by vcn to admit a polynomial compression. For this purpose, we first define the following problem. In Red-Blue Dominating Set, we are given a bipartite graph G with a vertex bipartition $V(G) = T \cup N$ and a non-negative integer k. A set of vertices $N' \subseteq N$ is said to be an *RBDS* if each vertex in T has a neighbor in N'. The aim of Red-Blue Dominating Set is to decide whether there exists an *RBDS* of size at most k in G. We shall use the following result.

▶ **Proposition 28** ([23]). *Red-Blue Dominating Set parameterized by $|T| + k$ does not admit a polynomial compression, unless NP ⊆ coNP/poly.*

We show that CnR parameterized by vcn does not have a polynomial compression by developing a polynomial parameter transformation from Red-Blue Dominating Set parameterized by $|T| + k$ to CnR parameterized by vcn.

Bipartite Graphs with Large Degree and Girth. For our reduction, we borrow a construction by Fomin at al. [27] of bipartite graphs having high girth and high minimum degree, which they used to prove NP-hardness (and $W[2]$-hardness for the solution size k) of CnR. For positive integers p, q, and r, we can construct a bipartite graph $H(p, q, r)$ with rqp^2 edges and a bipartition (X, Y), with $|X| = |Y| = pq$. The set X is partitioned into sets $U_1, \ldots, U_p$, and the set Y is partitioned into sets $W_1, \ldots W_p$, with $|U_i| = |W_i| = q$. By $H_{i,j}$ we denote the subgraph of $H(p, q, r)$ induced by $U_i \cup W_j$, and by $\mathsf{deg}_{i,j}(z)$ we denote the degree of vertex z in $H_{i,j}$. Fomin et al. [27] provided the following construction:

▶ **Proposition 29** ([27]). *Let $q \geq 2p(r+1)\frac{(p(r+1)-1)^6-1}{(p(r+1)-1)^2-1}$. Then, we can construct $H(p, q, r)$ in time $\mathcal{O}(r \cdot q \cdot p^2)$ with the following properties.*
1. *The girth of $H(p, q, r)$ is at least 6.*
2. *For every vertex $z \in V(H_{i,j})$ and every $i, j \in [p]$, we have $r - 1 \leq \mathsf{deg}_{i,j}(z) \leq r + 1$.*

Polynomial Parameter Transformation. Let (G, k) be an instance of Red-Blue Dominating Set with $V(G) = T \cup N$. First, we construct a graph G' with $V(G') = T' \cup N'$ from G by introducing two new vertices, x and y, such that $T' = T \cup \{x\}$ and $N' = N \cup \{y\}$, and $E(G') = E(G) \cup \{xy\}$. We have the following observation.

▶ **Observation 30.** *G has an RBDS of size at most k if and only if G' has an RBDS of size at most $k + 1$. Moreover, any RBDS of G' contains y.*

Now, we present the main construction for our reduction. Denote the vertex set $V(T')$ by $\{v_1, v_2, \ldots, v_{p'}, x\}$. Moreover, let $p = p' + 1$, $\ell = k + 1$, $r = \ell + 2$, and $q = \lceil 2p(r+1)\frac{(p(r+1)-1)^6-1}{(p(r+1)-1)^2-1}\rceil$.

We construct $H(p, q, r)$ such that each of U_i and W_i, for $0 < i \leq p'$, contains q copies of vertex v_i, and each of U_p and W_p contains q copies of vertex x. Now, we obtain a graph G'' by adding one more set of vertices P to $H(p, q, r)$ such that $V(P) = V(N')$. Moreover, if there is an edge between a vertex $u \in N'$ and a vertex $v_i \in T'$, then we add an edge between u and every vertex of U_i, and also between u and every vertex of W_i. Similarly, we add an edge between y and every vertex of U_p, and between y and every vertex of W_p. Finally, we make the vertex y adjacent to every vertex of P. See Figure 2 for reference. For correctness, we have the following lemma.

▶ **Lemma 31.** *G' has an RBDS of size at most ℓ if and only if G'' is ℓ-copwin.*

Proof. First, we show that if G' has an *RBDS* of size ℓ, then ℓ cops have a winning strategy in G''. Let $S \subseteq N'$ be an RBDS in G' of size at most ℓ. The cops begin by choosing the vertices corresponding to S in P. Observe that the vertex y has to be present in S. Since vertex y dominates each vertex in P, the robber cannot safely enter a vertex in P. Additionally, due to the construction of G'', the vertices of S dominate each vertex in H. Hence, the robber cannot safely enter a vertex in H. Therefore, the robber will be captured in the first move of the cops.

Next, we show that if there is no *RBDS* of size ℓ in G', then ℓ cops do not have a winning strategy. We prove this by giving a winning strategy for the robber. First, we show that the robber can safely enter the graph. In the beginning, let there be $\ell_1 \leq \ell$ cops in P and $\ell_2 \leq \ell$ cops in H. Since there is no *RBDS* of size ℓ in G', for every placement of at most ℓ cops in P, there exists at least one pair of U_i and W_i such that no vertex of U_i and W_i is dominated by the cops from P. Let U_i and W_i be one such pair of sets such that no vertex of U_i and W_i is dominated by the cops from P. Moreover, since each vertex of H can dominate at

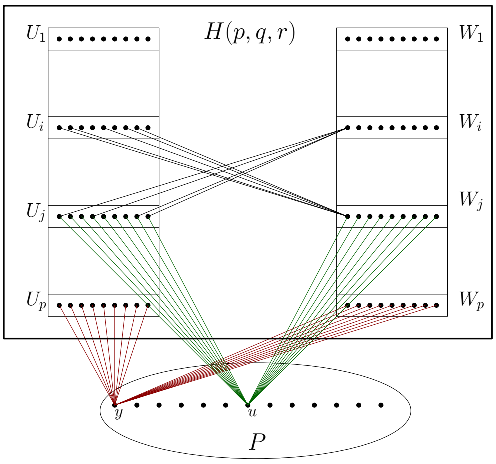

Figure 2 Illustration for $H(p, q, r)$ and P. If a vertex u is connected to v_j in G, then u is connected to every vertex of $W_j \cup U_j$. Moreover, for every i, j, each vertex in U_i has at least $r-1$ neighbors in U_j.

most $p(r+1)$ vertices in H, ℓ_2 cops can dominate at most $\ell \cdot p(r+1)$ vertices. Since U_i (and W_i also) contains q vertices, and $q > \ell \cdot p(r+1)$, the ℓ_2 cops in H cannot dominate all vertices of U_i, and hence the robber can safely enter a vertex of U_i.

Now, whenever the robber is under attack, it does the following. Without loss of generality, let us assume that the robber is in U_i (the case of W_i is symmetric). Since there are at most ℓ cops in P, there is always a W_j such that no vertex of W_j is dominated by cops from P. Since each vertex in U_i has at least $r-1 = \ell+1$ neighbours in W_j, the robber can move to at least $\ell+1$ vertices of W_j. Since the girth of H is at least 6, no vertex from H can dominate two vertices of W_j that are adjacent to the robber; else, we get a cycle on four vertices. Hence, at most ℓ cops from H can dominate at most ℓ neighbors of the robber in W_j, and the robber has at least $\ell+1$ neighbors in W_j. Hence, the robber can move to a safe vertex in W_j. Since the graph H is symmetric, the robber can move safely from $W_{j'}$ to $W_{i'}$ also. The robber follows this strategy to avoid capture forever. ◀

Next, we have the following observation to show that there exists a vertex cover U of G'' such that $|U| = poly(|T|, k)$.

▶ **Observation 32.** *$V(H) \cup \{y\}$ is a vertex cover of G''. Therefore, the vertex cover number of G'' is at most $2 \cdot p \cdot q + 1 = 1 + 2p \cdot \lceil 2p(k+3)\frac{(p(k+3)-1)^6-1}{(p(k+3)-1)^2-1} \rceil$, where $p = |T|+1$.*

This completes the proof of the argument that CNR parameterized by vcn is unlikely to admit a polynomial compression. Thus, we have the following theorem as a result of Lemma 31, Observation 32 and Proposition 28.

▶ **Theorem 4.** *CNR parameterized by* vcn *does not admit polynomial compression, unless NP ⊆ coNP/poly.*

6 Conclusion and Future Directions

To achieve our kernelization results, the rules we used concerned removing (false or true) twins from the graph. These rules are easy to implement and hence can be used to reduce the complexity of the input graph, even when the input graph is far from the considered parameters. For example, for cographs and grids, none of the considered parameters is constant/bounded, but cographs and grids can be reduced to a single vertex with the operation of removing twins, and hence, our reduction rules give an alternate proof that the cop number of cographs and grids is at most two [38, 19] for several variants. Moreover, MTP is well-studied with the motivation of designing computer games. Some examples of these variants include: multiple targets and multiple pursuer search [56] with applications in controlling non-player characters in video games; MTP from the robber's perspective with faster cops [44] where the strategies were tested on Baldur's Gate; MTP modeled with edge weights and different speeds of agents [36] with the specific examples of Company of Heroes and Supreme Commander. Moreover, the PACMAN game's movement can be considered as an instance of Fully Active CnR on a partial grid. One of the key aspects of designing these games is to come up with scenarios that are solvable but look complex and challenging. Our reduction rule can there. One can begin with an easy-to-resolve instance of CnR, and then keep adding twins to this instance (recursively) to get an instance that looks sufficiently complex but has the same complexity.

Finally, we define (formally defined in the full version [29]) a new variant of CnR, named Generalized CnR, that generalizes many well-studied variants of CnR. Here the input is $(G, \mathcal{C}_1, \ldots, \mathcal{C}_k, \mathcal{R})$ where each cop $\mathcal{C}_i$ has *speed* s_i and $\mathcal{R}$ has speed s_R. Moreover, each cop can be either forced to be *active* (have to move in each turn), *lazy* (at most one lazy cop moves in each turn), or *flexible*. Furthermore, each cop $\mathcal{C}_i$ can have *reach* λ_i. (Think of $\mathcal{C}_i$ having a gun with range λ_i, and if $\mathcal{R}$ is ever at a vertex that is at a distance at most λ_i from $\mathcal{C}_i$, it gets shot.) In the full version [29], we show that RR17 provides a kernel for Generalized CnR as well. This gives hope that RR17 can be used to get kernels for many practical variants not explicitly studied in this paper. Also, RR17 has been used to provide kernelization algorithm for Hunter and Rabbit game parameterized by vcn [22].

Still, many questions on the parameterized complexity of CnR remain open. We list some of these questions below.

▶ **Question 33.** *Does there exist an* FPT *algorithm for CnR parameterized by* vcn *with running time* $2^{\mathcal{O}(\mathsf{vcn})} \cdot n^{\mathcal{O}(1)}$*?*

▶ **Question 34.** *Does there exist a better bound for the cop number with respect to* vcn*? In particular, is* $\mathsf{c}(G) = o(\mathsf{vcn})$*?*

▶ **Question 35.** *Does CnR parameterized by* vcn *admit a polynomial* α*-approximate kernel?*

▶ **Question 36.** *Study CnR with respect to the following parameters: (1) feedback vertex set (2) treewidth (3) treedepth. In particular, is CnR* FPT *parameterized by treewidth?*

References

1. I. Abraham, C. Gavoille, A. Gupta, O. Neiman, and K. Talwar. Cops, robbers, and threatening skeletons: Padded decomposition for minor-free graphs. *SIAM Journal on Computing*, 48(3):1120–1145, 2019.
2. M. Aigner and M. Fromme. A game of cops and robbers. *Discrete Applied Mathematics*, 8(1):1–12, 1984.
3. T. Andreae. On a pursuit game played on graphs for which a minor is excluded. *Journal of Combinatorial Theory Series B*, 41(1):37–47, 1986.
4. D. Angelo, A. Navarra, and N. Nisse. A unified approach for gathering and exclusive searching on rings under weak assumptions. *Distributed Computing*, 30:17–48, 2017.

5 J. A. Baier, A. Botea, D. Harabor, and C. Hernández. Fast algorithm for catching a prey quickly in known and partially known game maps. *IEEE Transactions on Computational Intelligence and AI in Games*, 7(2):193–199, 2015. doi:10.1109/TCIAIG.2014.2337889.

6 A. Berarducci and B. Intrigila. On the cop number of a graph. *Advances in Applied mathematics*, 14(4):389–403, 1993.

7 A. Beveridge, A. Dudek, A. Frieze, and T. Müller. Cops and robbers on geometric graphs. *Combinatorics, Probability and Computing*, 21(6):816–834, 2012.

8 A. Bonato, S. Finbow, P. Gordinowicz, A. Haidar, W. B. Kinnersley, D. Mitsche, P. Prałat, and L. Stacho. The robber strikes back. In *proceedings of Computational Intelligence, Cyber Security and Computational Models*, volume 246, pages 3–12. Springer, 2014.

9 A. Bonato and R. Nowakowski. *The Game of Cops and Robbers on Graphs.* American Mathematical Society, 2011.

10 E. Bonnet, S. Gaspers, A. Lambilliotte, S. Rümmele, and A. Saffidine. The parameterized complexity of positional games. In *Proceedings of the 44th International Colloquium on Automata, Languages, and Programming (ICALP 2017)*, pages 90:1–90:14, 2017.

11 E. Bonnet, F. Jamain, and A. Saffidine. On the complexity of connection games. *Theoretical Computer Science*, 644:2–28, 2016.

12 N. Bowler, J. Erde, F. Lehner, and M. Pitz. Bounding the cop number of a graph by its genus. *SIAM Journal on Discrete Mathematics*, 35(4):2459–2489, 2021.

13 P. Bradshaw, S. A. Hosseini, and J. Turcotte. Cops and robbers on directed and undirected abelian Cayley graphs. *European Journal of Combinatorics*, 97, 2021.

14 S. Brandt, S. Pettie, and S. Uitto. Fine-grained lower bounds on cops and robbers. In Yossi Azar, Hannah Bast, and Grzegorz Herman, editors, *26th Annual European Symposium on Algorithms (ESA 2018)*, volume 112 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 9:1–9:12, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.ESA.2018.9.

15 A. C. Burgess, R. A. Cameron, N. E. Clarke, P. Danziger, S. Finbow, C. W. Jones, and D. A. Pike. Cops that surround a robber. *Discrete Applied Mathematics*, 285:552–566, 2020.

16 T. H. Chung, G. A. Hollinger, and V. Isler. Search and pursuit-evasion in mobile robotics: A survey. *Autonomous Robots*, 31(299):299–316, 2011.

17 M. Cygan, F. V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. *Parameterized Algorithms.* Springer Publishing Company, Incorporated, 1st edition, 2015.

18 J. Czyzowicz, L. Gasieniec, T. Gorry, E. Kranakis, R. Martin, and D. Pajak. Evacuating robots via unknown exit in a disk. In F. Kuhn, editor, *Distributed Computing (DISC 2014)*, volume 8784 of *Lecture Notes in Computer Science*, pages 122–136, 2018. doi:10.1007/978-3-662-45174-8_9.

19 S. Das and H. Gahlawat. Variations of cops and robbers game on grids. *Discrete Applied Mathematics*, 305:340–349, 2021.

20 S. Das and H. Gahlawat. On the Cop Number of String Graphs. In Sang Won Bae and Heejin Park, editors, *33rd International Symposium on Algorithms and Computation (ISAAC 2022)*, volume 248 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 45:1–45:18, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ISAAC.2022.45.

21 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.

22 T. Dissaux, Fioravantes F., H. Gahlawat, and N. Nicolas. Recontamination helps a lot to hunt a rabbit. In Bérôme Leroux, Sylvain Lombardy, and David Peleg, editors, *48th International Symposium on Mathematical Foundations of Computer Science (MFCS 2023)*, Leibniz International Proceedings in Informatics (LIPIcs), pages 40:1–40:14, Dagstuhl, Germany, 2023.

23 M. Dom, D. Lokshtanov, and S. Saurabh. Incompressibility through colors and ids. In *Proceedings of the 36th International Colloquium on Automata, Languages and Programming*, ICALP '09, pages 378–389, Berlin, Heidelberg, 2009. Springer-Verlag. doi:10.1007/978-3-642-02927-1_32.

24 H. E. Dudeney. *Amusements in Mathematics*. Facsimile Publisher, 1917.

25 J. A. Ellis, I. H. Sudborough, and J. S. Turner. The vertex separation and search number of a graph. *Information and Computation*, 113(1):50–79, 1994.

26 M. R. Fellows. The lost continent of polynomial time: Preprocessing and kernelization. In *Parameterized and Exact Computation, Second International Workshop, IWPEC 2006, Zürich, Switzerland, September 13-15, 2006, Proceedings*, IWPEC'06, pages 276–277, Berlin, Heidelberg, 2006. Springer-Verlag. doi:10.1007/11847250_25.

27 F. Fomin, P. Golovach, J. Kratochvíl, N. Nisse, and K. Suchan. Pursuing a fast robber on a graph. *Theoretical Computer Science*, 41(7-9):1167–1181, 2010.

28 F. V. Fomin, D. Lokshtanov, S. Saurabh, and M. Zehavi. *Kernelization: Theory of Parameterized Preprocessing*. Cambridge University Press, 2019. doi:10.1017/9781107415157.

29 H. Gahlawat and M. Zehavi. Parameterized analysis of the cops and robber problem. *arXiv:2307.04594*, 2023. doi:10.48550/arXiv.2307.04594.

30 N. Galesi, N. Talebanfard, and J. Torán. Cops-robber games and the resolution of tseitin formulas. *ACM Transactions on Computation Theory*, 12(2):1–22, 2020.

31 G. Gottlob, N. Leone, and F. Scarcello. A comparison of structural csp decomposition methods. *Artificial Intelligence*, 124(2):243–282, 2000.

32 G. Gottlob, N. Leone, and F. Scarcello. The complexity of acyclic conjunctive queries. *Journal of the ACM*, 48(3):431–498, 2001.

33 G. Gottlob, N. Leone, and F. Scarcello. Robbers, marshals, and guards: game theoretic and logical characterizations of hypertree width. *Journal of Computer and System Sciences*, 66(4):775–808, 2003.

34 I. Gromovikov, W. B. Kinnersleyb, and B. Seamone. Fully active cops and robbers. *Australian Journal of Combinatorics*, 76(2):248–265, 2020.

35 R. Isaacs. *Differential Games: A Mathematical Theory with Applications to Warfare and Pursuit, Control and Optimization*. John Wiley and Sons, New York, 1965.

36 A. Isaza, J. Lu, V. Bulitko, and R. Greiner. A cover-based approach to multi-agent moving target pursuit. In *proceedings of the Fourth Artificial Intelligence and Interactive Digital Entertainment Conference*, pages 54–59. AAAI Press, 2008.

37 T. Ishida and R. E. Korf. Moving target search. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence*, IJCAI'91, pages 204–210, San Francisco, CA, USA, 1991. Morgan Kaufmann Publishers Inc.

38 G. Joret, M. Kaminski, and D. O. Theis. Cops and robber game on graphs with forbidden (induced) subgraphs. *Contributions to Discrete Mathematics*, 5(2):40–51, 2010.

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

40 K. Klein and S. Suri. Catch me if you can: Pursuit and capture in polygonal environments with obstacles. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI 2012)*, volume 26, pages 2010–2016, 2012. doi:DOI:10.1609/aaai.v26i1.8375.

41 F. Lehner. On the cop number of toroidal graphs. *Journal of Combinatorial Theory, Series B*, 151:250–262, 2021.

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

43 M. Mamino. On the computational complexity of a game of cops and robbers. *Theoretical Computer Science*, 477:48–56, 2013.

44 C. Moldenhauer and N. R. Sturtevant. Evaluating strategies for running from the cops. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, IJCAI'09, pages 584–589, San Francisco, CA, USA, 2009. Morgan Kaufmann Publishers Inc.

45 N. Nisse. Network decontamination. In *Distributed Computing by Mobile Entities, Current Research in Moving and Computing*, volume 11340 of *Lecture Notes in Computer Science*, pages 516–548. Springer, 2019.

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

47 D. Offner and K. Ojakian. Variations of cops and robber on the hypercube. *Australasian Journal of Combinatorics*, 59(2):229–250, 2014.

48 T. D. Parsons. Pursuit-evasion in a graph. In *Theory and Applications of Graphs, LNCS*, volume 642, pages 426–441. Springer, 1978.

49 V. Patsko, S. Kumkov, and V. Turova. Pursuit-evasion games. In *Handbook of Dynamic Game Theory*, pages 951–1038. Springer, 2018.

50 J. Petr, J. Portier, and L. Versteegen. A faster algorithm for cops and robbers. *arXiv*, 2021. `arXiv:2112.07449v2`.

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

52 A. Scott. *On the Parameterized Complexity of Finding Short Winning Strategies in Combinatorial Games*. PhD thesis, University of Victoria, 2009.

53 A. Scott and U. Stege. Parameterized pursuit-evasion games. *Theoretical Computer Science*, 411(43):3845–3858, 2010.

54 P. D. Seymour and R. Thomas. Graph searching and a min-max theorem for tree-width. *Journal of Combinatorial Theory Series B*, 58(1):22–33, 1993.

55 D. P. Williamson and D. B. Shmoys. *The Design of Approximation Algorithms*. Cambridge University Press, 2011. URL: `http://www.cambridge.org/de/knowledge/isbn/item5759340/?site_locale=de_DE`.

56 F. Xie, A. Botea, and A. Kishimoto. A scalable approach to chasing multiple moving targets with multiple agents. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, IJCAI'17, pages 4470–4476. AAAI Press, 2017.