Fine-grained Lower Bounds on Cops and Robbers

Sebastian Brandt
ETH Zürich
Zürich, Switzerland
brandts@ethz.ch

Seth Pettie[1]
University of Michigan
Ann Arbor, MI, USA
pettie@umich.edu

Jara Uitto[2]
ETH Zürich
Zürich, Switzerland
jara.uitto@inf.ethz.ch

Abstract

Cops and Robbers is a classic pursuit-evasion game played between a group of g cops and one robber on an undirected N-vertex graph G. We prove that the complexity of deciding the winner in the game under optimal play requires $\Omega\left(N^{g-o(1)}\right)$ time on instances with $O(N\log^2 N)$ edges, conditioned on the Strong Exponential Time Hypothesis. Moreover, the problem of calculating the minimum number of cops needed to win the game is $2^{\Omega(\sqrt{N})}$, conditioned on the weaker Exponential Time Hypothesis. Our conditional lower bound comes very close to a conditional *upper bound*: if Meyniel's conjecture holds then the cop number can be decided in $2^{O(\sqrt{N}\log N)}$ time.

In recent years, the Strong Exponential Time Hypothesis has been used to obtain many lower bounds on classic combinatorial problems, such as graph diameter, LCS, EDIT-DISTANCE, and REGEXP matching. To our knowledge, these are the first conditional (S)ETH-hard lower bounds on a strategic game.

2012 ACM Subject Classification Mathematics of computing → Graph theory

Keywords and phrases Cops and Robbers

Digital Object Identifier 10.4230/LIPIcs.ESA.2018.9

1 Introduction

The game of *Cops and Robbers* is a two-player perfect information game played on a graph. One player is the *cop player*, who is identified with a set of g cops[3] occupying vertices of the graph. The other player is the *robber player*, who is identified with a single robber occupying some vertex of the graph. The game begins by the cop player placing the set of cops on the graph. Once she has decided the locations of the cops, it is the turn of the robber player to do the same.

Then, taking turns and initiated by the cop player, the players are allowed to move their pieces along the edges of the graph, where a turn of a player consists of moving all pieces the

[1] Supported by NSF grants CCF-1514383 and CCF-1637546.
[2] Partially supported by ERC Grant No. 336495 (ACDC).

[3] We decided to denote the number of cops by g as opposed to the "standard" k to avoid confusion later with the parameter k for k-CNF.

26th Annual European Symposium on Algorithms (ESA 2018).
Editors: Yossi Azar, Hannah Bast, and Grzegorz Herman; Article No. 9; pp. 9:1–9:12
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

player is identified with to an adjacent vertex. We assume that the graph is reflexive, i.e., a player is allowed to let a piece stay in the vertex it is currently occupying. The goal of the cop player is to *capture* the robber, i.e., move at least one cop to the vertex occupied by the robber. Conversely, the goal of the robber is to avoid being captured indefinitely. We say that a graph G is *g-cop-win* if there is a strategy for g cops to guarantee capture of the robber. Furthermore, we call the smallest integer g such that G is a g-cop-win graph the *cop number* of G and denote it by $c(G)$. Notice that any graph with n vertices is n-cop-win.

In this paper, we study the computational complexity of determining the cop number of a given input graph. It is known from previous work by Berarducci and Intrigila [5] that, for a *fixed* g, one can check in polynomial time whether $c(G) \leq g$. On the other hand, it was recently shown by Kinnersley that, for a *non-fixed* g, i.e., that can be a function of n, deciding whether $c(G) \leq g$ is EXPTIME-complete [22].

Perhaps the most famous and intriguing problem in the field of cops and robbers is Meyniel's conjecture, that states that $O(\sqrt{n})$ cops *always* suffice to capture the robber in any n-vertex graph [17]. Towards proving this conjecture, it is known that there exist graphs with cop number $\Theta(\sqrt{n})$ [26], and that $n/2^{(1+o(1))\sqrt{\log n}}$ cops always suffice to capture the robber; see Scott and Sudakov [30], or Lu and Peng [24] for a similar bound. Combining this upper bound with an $n^{O(g)}$ algorithm for checking whether the cop number is at most g [5], the cop number can always be computed in $n^{n/2^{(1+o(1))\sqrt{\log n}}}$ time. Moreover, assuming that Meyniel's conjecture is true, this upper bound reduces to $n^{O(\sqrt{n})}$. Hence, under this assumption $2^{\Omega(\sqrt{n}\log n)}$ is the best lower bound that we can hope to achieve.

But how close to this bound is it possible to get? While the result by Kinnersley shows EXPTIME-completeness, it gives relatively loose guarantees on the actual value in the exponent of the runtime. Since the completeness proof goes through a series of reductions [22, 31] and the size of the input graph grows (polynomially) in these reductions, the lower bound by Kinnersley "only" gives a $2^{n^{1/5}}$ lower bound.[4]

Our work can be seen as a step towards finding the right asymptotic bound in the exponent. Furthermore, our construction is quite simple and, in particular, gives rise to very concise and easy to understand strategies for the players. To state our main results, we recall the satisfiability problem and the definitions of the exponential time hypotheses below.

▶ **Definition 1.** Let c_k be the smallest value such that instances of k-CNF-SAT with m clauses and n variables can be solved in $2^{(c_k+o(1))n}\,\mathrm{poly}(m)$ time. The *Exponential Time Hypothesis* (ETH) is that $c_k > 0$ for all $k \geq 3$. The *Strong Exponential Time Hypothesis* is that $\lim_{k\to\infty} c_k = 1$, i.e., k-CNF-SAT requires $2^{(1-o(1))n}$ time for any non-constant $k = k(n)$.

Conditioning on the Exponential Time Hypothesis and the Strong Exponential Time Hypothesis, we prove the following theorems. We want to emphasize that Theorem 2 is optimal up to a constant factor in the exponent and Theorem 3 and Theorem 4 are optimal up to a $\log N$ factor in the exponent, in the case of Theorem 3 under the assumption of Meyniel's conjecture. Furthermore, a potentially interesting detail of Theorem 2 is that it works for *any* $g \geq 2$, i.e., not only when g grows large.

[4] Suppose an ABF game [31] is played on a CNF formula with ℓ variables and $O(\ell)$ clauses. Kinnersley [22] reduces this to a *lazy cops and robbers with protection* game on $O(\ell^2)$ vertices, $O(\ell^3)$ edges, and $\ell + O(1)$ cops. Given any such game with n vertices, m edges, and g cops, Kinnersley [22] reduces it to an equivalent cops and robbers with protection game on $O(gn+m)$ vertices, $O(n(g^2+m))$ edges, and g cops. Mamino's reduction [25] from cops and robbers with protection to standard cops and robbers transforms a game with parameters n, m, g to $O(g^2 n), O(g^4 m), g$. Composing all three reductions, we arrive at a standard cops and robbers instance with $N = O(\ell^5)$ vertices, $O(\ell^9)$ edges, and $\ell + O(1)$ cops. If we need $2^{\Omega(\ell)}$ time to decide the winner of the original ABF game, then this gives, at the best, a $2^{\Omega(N^{1/5})}$ lower bound on deciding the cop number of an N-vertex graph.

▶ **Theorem 2.** *Fix an integer $g \geq 2$ and any $\delta > 0$. Conditioned on the Strong Exponential Time Hypothesis, the problem of deciding whether an N-vertex, M-edge graph has cop number at most g cannot be solved in $O(M^{g-\delta})$ time.*

In an informal sense, Theorem 2 can be interpreted as the statement that exploring almost all of the $O(M^{g+1})$, resp. $O(M^{g+2})$, possible game configurations and transitions between these configurations in a cops and robbers game with g, resp. $g+1$, cops is unavoidable in order to determine whether the cop number is at most g or at least $g+1$.

▶ **Theorem 3.** *Conditioned on the Exponential Time Hypothesis, the problem of calculating the cop number of an N-vertex graph cannot be solved in $2^{o(\sqrt{N})}$ time.*

As mentioned above, if Meyniel's conjecture is true, the lower bound given in Theorem 3 cannot be improved by more than a log-factor in the exponent. However, if Meyniel's conjecture turns out to be false and there is an infinite graph family requiring $\Omega(X(N))$ cops to capture the robber, for some function $X(N) = \omega(\sqrt{N})$, there is well-founded hope that our approach can be used to show that the problem of calculating the cop number of an N-vertex cannot be solved in $2^{o(X(n))}$, thereby staying in the realm of being only a log-factor away from the optimum. The reason for this hope is that the graphs we construct in order to infer our lower bound contain components that are essentially the hard instances for the $\Omega(\sqrt{N})$ lower bound on the cop number. Of course, it cannot be taken for granted that all proof details still work out if we replace these components with the hard instances for a larger lower bound on the cop number, but the simplicity of our construction suggests that this might indeed be the case.

▶ **Theorem 4.** *Let $g : \mathbb{N} \to \mathbb{R}$ be any function such that $g(x) = o(\sqrt{x})$ and $g(x+1) \leq g(x)+1$ for all positive integers x. Conditioned on the Exponential Time Hypothesis, the problem of deciding whether the cop number of an N-vertex graph is at most $g(N)$ cannot be solved in $2^{o(g(N))}$ time.*

Informally, Theorem 4 states that also for all ("natural") functions between constant functions and $\Theta(\sqrt{N})$, deciding whether the cop number of a graph is bounded by the function takes time exponential in the function. Similarly to the case of Theorem 3, in case Meyniel's conjecture turns out to be false, the range of functions for which Theorem 4 applies might be increased to include functions from $\omega(\sqrt{N})$ by adapting our graph construction in a straightforward way.

To the best of our knowledge, this is the first work to apply the (Strong) Exponential Time Hypothesis on a strategic game. In previous works, (S)ETH has been applied to, e.g., some well known combinatorial problems such as graph diameter [29], LCS [9, 10], EDIT-DISTANCE [3], and REGEXP matching [4, 11].

2 Related Work

The study of the game of Cops and Robbers was initiated by Quilliot [27] in 1978 and introduced independently a few years later by Nowakowski and Winkler [7]. Nowakowski and Winkler provided a full characterization of graphs where one cop can capture a robber which was later extended to the case of many cops by Clarke and MacGillivray [14]. One of the core questions related to the game is the *cop number* of a graph, which denotes the minimum number of cops required to capture the robber. A very early result by Aigner and Fromme states that 3 cops suffice to capture a robber on planar graphs and in the same work, they showed that any graph with girth at least 5 and minimum degree at least δ has a cop number of at least δ [2].

Later, Prałat showed that there are incidence graphs of projective planes that satisfy these properties for $\delta = \Omega(\sqrt{n})$ yielding the state-of-the-art lower bound for the cop number of any graph [26]. Given that Meyniel's [17] conjectured $O(\sqrt{n})$ upper bound holds, this bound is tight. The current best upper bound of $n/2^{(1+o(1))\sqrt{\log n}}$ [30] is far away from this though and improving it is perhaps the most crucial open problem in the field.

Beyond the existential question of determining the maximum cop number, there is the computational question. On the positive side, for fixed g, determining whether the cop number is at most g can be computed in polynomial time [5]. To the best of our knowledge, the current best algorithm runs in $O(n^{2g+3})$ time [6]. Many years later, this was contrasted by a negative result showing that for a non-fixed g, i.e., g can be a function of the number of vertices n, this question becomes NP-hard [16]. A bit later, it was shown by Mamino that this question is hard for PSPACE [25]. An interesting detail on this work is that it goes through a reduction to a variant called "Cops and Robbers with Protection". In this variant, edges are divided into protected and unprotected edges. The crux of the game is that the capture only occurs if a cop moves to the vertex occupied by the robber through an unprotected edge. In a recent breakthrough, Kinnersley managed to show that the stardard variant of the problem is actually EXPTIME-complete [22].

Even though the progress on the specific question of finding the cop number is fairly recent, other related questions in various graph classes have been studied long ago. For example, in the end of seventies and beginning of eighties, Adachi et al. studied a variant of the game where one cop is trying to prevent any of multiple robbers from reaching a "hole" in the graph [21, 1]. In their variant, the initial positions are fixed and the cop and exactly one robber have to move in each turn. They showed EXPTIME-completeness. For a survey of earlier complexity results, we refer to a survey by Johnson [20]

Goldstein and Reingold [18] studied a version of the game in which the cops and robbers have prescribed initial positions and the goal of the robber is to reach a specific vertex. They showed that in undirected graphs this variant of the game is EXPTIME-complete. In the same work, they showed that the directed version of the game, without fixing the initial positions, is also EXPTIME-complete.

For the curious reader, we point out that many of the results listed here are based on reductions to the ABF-problem that was shown to be EXPTIME-complete by Stockmeyer and Chandra [31]. Furthermore, for a great survey on the results of the game we refer the reader to the book by Bonato and Nowakowski [7].

3 Preliminaries

Let us give some definitions that are used throughout the paper.

▶ **Definition 5** (k-CNF-SAT). The input to the k-CNF-SAT problem is a conjunction of one or more clauses, where each clause consists of a disjunction of at most k literals. The goal is to determine whether the formula is *satisfiable*, i.e., if there is a truth assignment of the variables such that the input formula evaluates to true.

Especially, we wish to specify what we mean by a *partial assignment* of variables in a logical formula consisting only of literals, disjunctions, and conjunctions. In a partial assignment, a subset of the variables is set to true/false and some may be left unassigned. A disjunctive clause $x_1 \vee x_2 \vee \cdots \vee x_\ell$, for $\ell \geq 1$ is *satisfied* by a partial assignment if at least one literal in the clause has an assigned truth value and is true. We point out that this means that a disjunctive clause that contains both a variable and its negation can still be unsatisfied by a partial assignment. Throughout the paper, we denote the number of variables in a k-CNF-SAT instance by n, the number of vertices in a graph by N, the number of cops by g, and we reserve the letter k as the parameter for k-CNF-SAT.

4 The Construction

Fix a number $g \geq 2$ of cops and an integer $k \geq 3$. The technique we use to derive our main results is a reduction from k-CNF-SAT to the problem of deciding whether a graph is g-cop-win. To this end, we will start this section by describing how we transform any k-CNF formula with n variables and m clauses into an input graph for the Cops and Robbers game with g cops. Then we will prove that our graph construction has the property that the cops can win the game in the constructed graph iff the k-CNF formula is satisfiable. We will conclude the section by using this property to infer our lower bounds.

In the following we give an informal high-level overview of our construction. We say that vertex v *covers* a set of vertices S if v is adjacent to all vertices in S. Vertex v always covers itself. The constructed graph consists of two zones: one that is designed for the cops from which they can cover the whole graph if the k-CNF formula is satisfiable, and one for the robber in which he can evade capture indefinitely if the formula is unsatisfiable.

The cops' zone consists of $g2^{\lceil n/g \rceil}$ vertices, which represent certain partial assignments to groups of $\lceil n/g \rceil$ variables in the CNF formula. By occupying g non-conflicting partial assignments, the cops can collectively represent a total assignment to the variables. If this total assignment is satisfying, then it should cover *every* vertex in the robber area, leaving the robber nowhere to go. (Each vertex in the robber's zone is associated with a clause, which is covered by the cops if their collective assignment satisfies the clause.) On the other hand, if no satisfying assignment exists, then the robber must always be able to move to some vertex not covered by any cop.

If the cops and robbers agreed to stay in their own zones then the construction of the robber's zone could be very simple: m vertices (one for each clause) arranged in a clique suffices. Of course, both the robber and the cops are free to roam over the whole graph, so we need to add extra mechanisms to dissuade the robber from entering the cops' zone, and protect the robber against any cops entering the robber's zone. To protect the robber, we make the subgraph induced by the robber's zone a girth-6 graph,[5] which means that any cop that enters the robber's zone can never cover more than one neighbor of the robber, leaving many options for the robber to escape. The mechanism to dissuade the robber from entering the cops' zone is more subtle; it ensures that any robber that does this loses in two turns, regardless of whether the k-CNF formula is satisfiable or not.

Because we are interested in lower bounds *as a function of input size*, it is important to keep the graph as sparse as possible. Many transformations on cops and robbers games (e.g., [22, 23, 25]) create very dense graphs, sometimes having $\Omega(n^2)$ edges. Parts of our construction could be simplified by introducing large cliques, but this would weaken the resulting (conditional) lower bounds. This concludes the informal overview; in the following, we will give a formal description of our graph construction.

Let $\phi = C_1 \wedge \cdots \wedge C_m$ be a k-CNF formula over the variable set $\mathcal{V} = \{v_1, \ldots, v_n\}$, where the ith clause is $C_i = x_{i,1} \vee \cdots \vee x_{i,k}$ and each $x_{i,j}$ is a variable or its negation. The variable set is partitioned into $g \geq 2$ groups $\mathcal{V}_1, \ldots, \mathcal{V}_g$ of at most $\lceil n/g \rceil$ variables each. For reasons that will become clear later, it is desirable that the formula has the property that any *partial*

[5] "Girth" is the length of the shortest cycle.

$$\phi' \left\{ \begin{array}{l} \overset{C_1}{(v_1 \vee v_2 \vee v_3)} \wedge \overset{C_2}{(\neg v_3 \vee v_4)} \wedge \overset{C_3}{(v_2 \vee \neg v_4)} \\ \quad \wedge \, (v_1 \vee \neg v_1 \vee v_2 \vee \neg v_2) \wedge (v_3 \vee \neg v_3 \vee v_4 \vee \neg v_4) \end{array} \right. \begin{array}{l} \} \; \phi \\ \} \; C_4 \wedge C_5 \end{array}$$

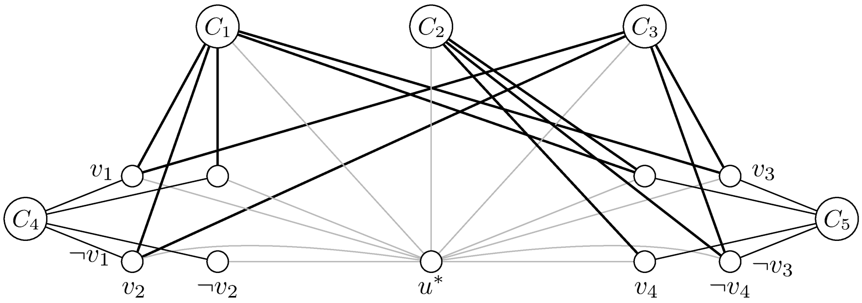


Figure 1 A schematic and simplified illustration of our graph construction in the case of two cops and a k-CNF formula ϕ with four variables and three clauses. Notice that vertex u^* is not connected to the two extra clauses C_4 and C_5. Each vertex in the figure labeled with C_i for some i corresponds to the set of vertices in B with clause-type i. The small vertices correspond to the partial truth assignments and are connected to the clause vertices that they satisfy (i.e., the corresponding literal is contained in these clauses). Notice that C_4 and C_5 are only covered by the cops if they occupy a set of vertices that corresponds to assigning a value to all variables. The edges between vertices in the C_i, i.e., between vertices in B, are not shown in Figure 1. For an illustration of these edges, see Figure 2.

satisfying truth assignment must set at least one variable in each group. To this end, we supplement ϕ with g extra clauses. Define ϕ' as follows.

$$\phi' = C_1 \wedge \cdots \wedge C_m \wedge C_{m+1} \wedge \cdots \wedge C_{m+g}$$
$$\text{where } C_{m+i} = \bigvee_{v \in \mathcal{V}_i} (v \vee \neg v)$$

Observe that ϕ' is satisfiable iff ϕ is since *any* total assignment to $\mathcal{V}$ automatically satisfies each of the clauses $C_{m+1}, \ldots, C_{m+g}$. Define $\overline{m} = m + g$.

The next step is to convert ϕ' to the graph G on which the Cops and Robbers game will be played. See Figure 1 for a simplified illustration of a graph constructed from a k-CNF-SAT instance.

Vertices

The vertex set $V(G)$ is $A_1 \cup \cdots \cup A_g \cup B \cup \{u^\star\}$, where there is a vertex $u \in A_i$ for each truth assignment $\psi_u : \mathcal{V}_i \to \{\mathsf{T}, \mathsf{F}\}$ to the ith variable group. The set B consists of $\Theta(\overline{m}^2)$ vertices, each of which is associated with one of the $\overline{m}$ clauses in ϕ'. If $u \in B$, clause$(u) \in [\overline{m}]$ indicates the clause index associated with u, and we say that u has *clause-type* clause(u). The role of $u^\star$ will be revealed shortly. In total, $|V(G)| = O(g2^{n/g} + \overline{m}^2)$.

Edges

The edge set $E(G)$ includes edges of three types:

Satisfaction Edges. Edges join partial assignments to clauses iff the partial assignment satisfies the clause:

$$\{\{u, u'\} \mid u \in A_i, u' \in B, \text{clause}(u') = q, \text{ and } \psi_u \text{ satisfies } C_q\} \subset E(G)$$

Special $u^\star$ Edges. In some ways, $u^\star$ functions like an assignment that magically satisfies all clauses C_i where $i \in [m]$, but none where $i \in [\overline{m}]\backslash[m]$. It is also adjacent to all vertices in $A_1, \ldots, A_g$.

$$\{\{u^\star, u\} \mid \text{Either } u \in A_1 \cup \cdots \cup A_g \text{ or } u \in B \text{ and clause}(u) \in [m]\} \subset E(G),$$

High Girth Subgraph. The subgraph of G induced by B has $\Theta(|B|^{3/2}) = \Theta(\overline{m}^3)$ edges and girth at least 6. Moreover, for each $u \in B$ and each $q \in [\overline{m}]$,

$$|\{u' \in B \mid \{u, u'\} \in E(G) \text{ and } \text{clause}(u') = q\}| \geq 1 \ .$$

I.e., each B-vertex has at least one neighbor of each clause-type.

It is not immediate from the description that the subgraph induced by B actually exists. We construct such a graph and clause-assignment now. Let p be the first prime greater than $\overline{m}$, so $p = \Theta(\overline{m})$, by Bertrand's postulate [13]. Define line(s,t) to be the line in $\mathbb{Z}_p^2$ with *slope* s and *offset* t:

$$\text{line}(s,t) = \{(i,j) \in \mathbb{Z}_p^2 \mid i \cdot s + t \equiv j \pmod p\}.$$

The set B consists of $2p^2$ vertices $\{w_{i,j}, l_{s,t} \mid (i,j),(s,t) \in \mathbb{Z}_p^2\}$, where $w_{i,j}$ represents the point (i,j) and $l_{s,t}$ represents line(s,t). The subgraph induced by B is simply the point-line incidence graph, i.e.,

$$\{w_{i,j}, l_{s,t}\} \in E(G) \Leftrightarrow (i,j) \in \text{line}(s,t) \ .$$

We restate some properties of this graph that were shown in previous work [8]. See [12, 15, 28, 33, 32, 26] for other constructions with essentially the same properties.

▶ **Lemma 6.** *Consider the p^2 points and p^2 lines indexed by $\mathbb{Z}_p^2$.*
1. *The intersection of two lines contains at most one point.*
2. *Two points are contained in at most one common line.*
3. *For any point (i,j) and any slope s, there exists some* line(s,t) *containing (i,j).*
4. *For any* line(s,t) *and index i, there is some point $(i,j) \in$* line(s,t).

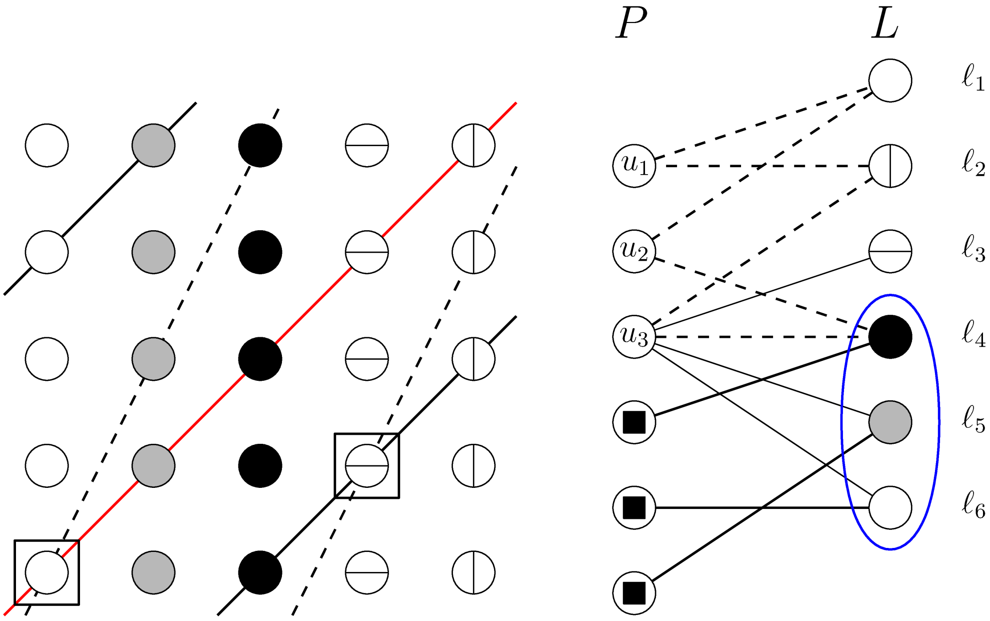
Properties (1) and (2) of Lemma 6 imply that the subgraph induced by B has no 4-cycles. Since it is clearly bipartite, it must have girth (at least) 6. We use properties (3) and (4) of Lemma 6 to design a good clause-assignment function clause : $B \to [\overline{m}]$. In particular,

$$\text{For points, } \text{clause}(w_{i,j}) = i + 1$$
$$\text{For lines, } \text{clause}(l_{s,t}) = s + 1$$

Since $p \geq \overline{m}$, it follows that for each clause index $q \in [\overline{m}]$, every point $w_{i,j}$ has at least one neighboring line with clause-type q, and every line $l_{s,t}$ has at least one neighboring point with clause-type q. See Figure 2 for an illustration.

This concludes the description of graph G. It is straightforward to construct G in time linear in the number of edges, which is $O(\overline{m}^2 g 2^{n/g} + \overline{m}^3)$. The following lemma shows that the construction indeed satisfies its purpose, i.e., the constructed graph G is g-cop-win if and only if the k-CNF formula ϕ is satisfiable.

▶ **Lemma 7.** *In the Cops and Robbers game on G with g cops, the cops have a winning strategy iff ϕ is satisfiable.*



(a) The subgraph B can be seen as a set of vertices/points and lines on a plane. In the figure, the points associated with the same clause are illustrated by the same color. Lines with slope 1 and offsets 0 and 3 are illustrated by solid red and black lines, respectively. Line with slope 2 and offset 0 is illustrated by a dashed line. The two unique intersections of the non-parallel lines are emphasized with black boxes.

(b) Concretely, the subgraph B is a bipartite graph with points on one side (left) and the lines on the other. Since every two lines have at most one intersection point, at most one neighbor of a point vertex u_3 can be covered by any other point vertex (see black boxes in the figure). Hence, 5 cops are needed to cover the neighbors of u_3. The same line of reasoning holds for any line vertex.

Figure 2 The subgraph depicted as a set of points and lines on a plane and as a bipartite graph. Notice that any cycle starting from a point vertex must pass through at least 3 line vertices. Therefore, the girth of the graph is at least 6. This is illustrated by the dashed edges incident on vertices u_1, u_2, and u_3 on the right. Notice that the pictures above are not inferred from each other.

Proof. Suppose $\psi : \mathcal{V} \to \{\mathsf{T}, \mathsf{F}\}$ is a satisfying total assignment, decomposed into partial assignments $\psi_{u_1}, \ldots, \psi_{u_g}$, where ψ_{u_i} is associated with $u_i \in A_i$. In their initial move, the cops position themselves on $u_1, \ldots, u_g$. At this point they cover all vertices in $B \cup \{u^\star\}$, but leave the remaining vertices in $A_1 \cup \cdots \cup A_g$ uncovered. Without loss of generality we can assume that the robber begins at a vertex in $A_1 \backslash \{u_1\}$. In the next move, the cops stay put, except for the cop on u_2, which moves to $u^\star$. At this point all B-vertices with clause-types in $[m]$ are covered by the cop on $u^\star$, and those with clause-type $m+1$ are covered by the cop on u_1. The robber, being in A_1, can move once more or stay put, but is immediately caught by the cop on u_1 or $u^\star$ in the next turn.

Now consider the case where ϕ is unsatisfiable. We show that the robber has a winning strategy such that it never leaves the set B. Consider any moment in the middle of the game, after the cops have moved to vertices $w_1, \ldots, w_g$. The robber is located at some $w' \in B$ and may be forced to move if w' is in the neighborhood of $w_1, \ldots, w_g$. Let $z \geq 0$ be the number of cops that are located at some vertex in B. First consider the case that at least $z+1$ of the A_i do not contain any cop, and without loss of generality, let $A_1, \ldots, A_{z+1}$ contain no cop. By the properties of G, the robber is adjacent to a set of vertices $S = \{w'_1, \ldots, w'_{z+1}\} \subset B$, where $\text{clause}(w'_i) = m + i$. None of the S-vertices are covered by the $g - z$ cops stationed in $A_{z+2} \cup \cdots \cup A_g \cup \{u^\star\}$. Since the subgraph induced by B has girth at least 6, each of the remaining z cops can cover at most one S-vertex, hence at least one S-vertex is not covered by any cop, and the robber can move there without being captured. Now consider the other case, i.e., that exactly z of the A_i, say $A_1, \ldots, A_z$, do not contain any cop. Then the $g - z$ sets $A_{z+1}, \ldots, A_g$ contain exactly one cop each. Assume without loss of generality that $w_{z+1} \in A_{z+1}, \ldots, w_g \in A_g$, and let ψ' be the partial assignment obtained by combining the partial assignments $\psi_{w_{z+1}}, \ldots, \psi_{w_g}$. Since ϕ is unsatisfiable, there is a clause from ϕ not satisfied by ψ', say clause C_q. Similarly to the previous case, by the properties of G, the robber is adjacent to a set of $z+1$ vertices $S = \{w'_1, \ldots, w'_z, w'\} \subset B$, where clause$(w'_i) = m + i$ and clause$(w') = q$. Again, none of the S-vertices is covered by the $g - z$ cops stationed in $A_{z+1} \cup \cdots \cup A_g$ and the remaining z cops can cover at most z S-vertices. Now, with the same argumentation as in the previous case, it follows that there is an S-vertex the robber can move to without being captured.

The arguments above apply to any stage in the middle of the game; the same arguments show that if ϕ is unsatisfiable, the robber has a safe *first* move, after the cops choose their initial positions. ◀

We also obtain the following curious observation from our construction. Later, we use the observation to slightly strengthen our results, but we also believe that it is a property of the construction that is of independent interest.

▶ **Observation 8.** *Recall the vertex set of G is $V(G) = A_1 \cup \cdots \cup A_g \cup B \cup \{u^\star\}$. Then the cop number of G is either g or $g+1$.*

Proof. If there are $g+1$ cops, they can position themselves on vertices $u_1, \ldots, u_g, u^\star$ with $u_i \in A_i$. Then, since $u^\star$ is connected to all vertices in $A_1 \cup \cdots \cup A_g \cup B$ except those in B with clause-type in $[\overline{m}] \backslash [m]$, and for each i, the cop in u_i covers all vertices in B with clause-type $m + i$, the cops cover the entire graph and hence can capture the robber in the following turn. If, on the other hand, there are at most $g - 1$ cops, then the robber has a simple winning strategy by always moving to a vertex in B with clause-type in $[\overline{m}] \backslash [m]$ that is not covered by any cop. By analogous arguments to the ones used in the proof of Lemma 7, such a vertex always exists. ◀

5 Hardness of Finding the Cop Number

Quickly before going into the proofs of our main theorems, we point out a small technical detail. The input k-CNF-SAT instance that we reduce to the Cops and Robbers instance may contain a very large number of clauses. This would then imply that our graph constuction has many edges, up to around $\hat{m}^4$ edges, where $\hat{m}$ is the number of input clauses. This would in turn result in a running time for our construction that is too large for our purposes. We can work around this problem by using the sparsification lemma [19], which, for any chosen $\epsilon > 0$, reduces an arbitrary k-CNF-SAT instance to $2^{\epsilon n}$ k-CNF-SAT instances with at most $c(k, \epsilon) \cdot n$ clauses each, where $c(k, \epsilon)$ is a function independent of n.

Next, we prove Theorem 2, i.e., that under the Strong Exponential Time Hypothesis, the time needed to decide whether the cop number is at most some fixed g grows exponentially as a function of g. A proof sketch goes as follows. We are given a k-CNF-SAT instance with n variables and $O(n)$ clauses. We obtain a graph with roughly $2^{n/g}$ vertices and edges from our construction. Being able to solve our cop number decision problem in $M^{g-\delta} = \left(2^{n/g}\right)^{g-\delta} \ll 2^n$ time yields a contradiction to the Strong Exponential Time Hypothesis.

▶ **Theorem 2.** *Fix an integer $g \geq 2$ and any $\delta > 0$. Conditioned on the Strong Exponential Time Hypothesis, the problem of deciding whether an N-vertex, M-edge graph has cop number at most g cannot be solved in $O(M^{g-\delta})$ time.*

Proof. Let $\hat{\phi}$ be an instance of k-CNF-SAT with $\hat{m}$ clauses and n variables, and let $\epsilon > 0$. Using the sparsification lemma, in $\text{poly}(n) \cdot 2^{\epsilon n}$ time we can reduce $\hat{\phi}$ to $2^{\epsilon n}$ instances of k-CNF-SAT, each having at most $m = c(k, \epsilon) \cdot n$ clauses. Let ϕ be one of those instances, and let G be the graph obtained by applying our graph construction to ϕ. G is an N-vertex, M-edge graph, where $N = \Theta(g2^{n/g} + m^2) = \Theta(2^{n/g})$ and $M = O(m^2 \cdot N) = O(N \log^2 N)$. Thus, if we can decide in $O(M^{g-\delta}) = O(\text{poly}(m)N^{g-\delta})$ time whether G has cop number g, we can determine the satisfiability of $\hat{\phi}$ in $\text{poly}(m)2^{\epsilon n} \cdot N^{g-\delta} = \text{poly}(n)2^{n(\epsilon+1-\delta/g)}$ time, by Lemma 7. The calculations above do not depend on the value of k, so setting $\epsilon < \delta/g$ contradicts the Strong Exponential Time Hypothesis. ◀

Next, we provide the proof for Theorem 3. We note that this result can also be obtained from extending the proof of Theorem 4 to functions $g(x) = \Theta(x)$ (which requires some extra care), but this special case is much cleaner to prove and has all the same ingredients. The main difference is in the simplicity of calculations. The difference to the proof of Theorem 2 is that since g is a function of n, we can set $g = n$ and the graph becomes much smaller in terms of the number of variables of the input k-CNF formula. As a consequence, the dominating part of the constructed graph G w.r.t size is now B (and not the A_i, as in the proof of Theorem 2).

▶ **Theorem 3.** *Conditioned on the Exponential Time Hypothesis, the problem of calculating the cop number of an N-vertex graph cannot be solved in $2^{o(\sqrt{N})}$ time.*

Proof. Fix an arbitrarily small constant ϵ and an integer $k \geq 3$. Let $\hat{\phi}$ be an instance of k-CNF-SAT with $\hat{m}$ clauses and n variables, and ϕ be one of the $2^{\epsilon n}$ instances with $m = c(k, \epsilon)n$ clauses generated from the sparsification lemma. Use our graph construction to create a graph G from ϕ for a Cops and Robbers game with $g = n$ cops. G has $N = \Theta(g2^{n/g} + (n + m)^2) = \Theta(m^2)$ vertices and $O(m^2 g 2^{n/g} + m^3) = O(m^3)$ edges. If we can determine the cop number of G in $2^{o(\sqrt{N})} = 2^{o(m)} = 2^{o(n)}$ time, we can determine the satisfiability of $\hat{\phi}$ in $\text{poly}(n)2^{\epsilon n} \cdot 2^{o(n)} = \text{poly}(n)2^{(\epsilon+o(1))n}$ time, by Lemma 7. Since ϵ can be made arbitrarily small, this contradicts the Exponential Time Hypothesis. ◀

As our last technical contribution, we show that one can replace the $\sqrt{N}$ in the exponent in Theorem 3 with essentially any reasonable function in N that is asymptotically smaller than $\sqrt{N}$ and obtain a lower bound for deciding whether the cop number of an input graph is bounded by this function. Basically, the statement of this theorem, combined with Observation 8 is that even when we know that the cop number is either $g(N)$ or $g(N) + 1$, the decision problem is hard.

▶ **Theorem 4.** *Let $g : \mathbb{N} \to \mathbb{R}$ be any function such that $g(x) = o(\sqrt{x})$ and $g(x+1) \leq g(x)+1$ for all positive integers x. Conditioned on the Exponential Time Hypothesis, the problem of deciding whether the cop number of an N-vertex graph is at most $g(N)$ cannot be solved in $2^{o(g(N))}$ time.*

Proof. Let ϵ, k, $\hat{\phi}$, $\hat{m}$, n, ϕ and m be as in the proof of Theorem 3. Use our graph construction to create a graph G from ϕ for a Cops and Robbers game with n cops and denote the number of vertices of G by N. Check whether $g(N) < n + 1$. Observe that since $g(x) = o(\sqrt{x})$ and $N = O(n^2)$, there is some constant n_0 such that $g(N) < n + 1$ for all possible k-CNF formulae ϕ with $n \geq n_0$ variables. Hence, if $g(N) \geq n + 1$, n is constant and we can decide in constant time whether ϕ is satisfiable.

Now consider the other case, i.e., $g(N) < n+1$. As we want to use the Exponential Time Hypothesis in order to infer a conditional lower bound on the time it takes to determine whether the cop number of an N-vertex graph is at most $g(N)$, we would like the constructed graph G to have the property that ϕ is satisfiable iff G has cop number at most $g(N)$. With the current construction of G we only have a similar property, namely, that ϕ is satisfiable iff G has cop number at most n, due to Lemma 7. But since $g(N) < n+1$, we can change G slightly, adding more and more vertices to G in a way that does not change the cop number of G, and in the end obtain a graph G' with the desired property. More specifically, we obtain G' from G by appending a path of r vertices to some arbitrarily chosen vertex u of G, where r is the smallest non-negative integer such that $g(N+r) \geq n$.

Set $N' = N + r$. Due to the properties of our function g, we know that $g(N') < n+1$. Note that the cop number of G' is the same as the cop number of G: In the case that ϕ is unsatisfiable, our robber strategy still works with the same arguments as in G. In the case that ϕ is satisfiable, the cops can simply perform the same strategy as in G, where they assume that the robber is in u if the robber is actually in one of the newly appended vertices. With this strategy, after 2 turns per player, the cops have captured the robber or at least one cop ends up at u while the robber is in one of the new vertices, in which case the robber can be captured by letting the cop traverse the appended path to the other end.

From the discussion above it follows that ϕ is satisfiable iff G' has cop number at most $g(N')$. Hence, if the problem of deciding whether the cop number of an N-vertex graph is at most $g(N)$ can be solved in $2^{o(g(N))} = 2^{o(n)}$ time, we can determine the satisfiability of $\hat{\phi}$ in $\text{poly}(n)2^{\epsilon n} \cdot 2^{o(n)} = \text{poly}(n)2^{(\epsilon + o(1))n}$ time. Since ϵ can be made arbitrarily small, this contradicts the Exponential Time Hypothesis. ◀

References

1 Akeo Adachi, Shigeki Iwata, and Takumi Kasai. Some combinatorial game problems require $\omega(n^k)$ time. *J. ACM*, 31(2):361–376, 1984.

2 Martin Aigner and Michael Fromme. A game of cops and robbers. *Discrete Applied Mathematics*, 8:1–12, 1984.

3 A. Backurs and P. Indyk. Edit distance cannot be computed in strongly subquadratic time (unless SETH is false). In *Proceedings 47th Annual ACM Symposium on Theory of Computing (STOC)*, pages 51–58, 2015.

4 Arturs Backurs and Piotr Indyk. Which regular expression patterns are hard to match? In *Proceedings of the 57th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 457–466, 2016. doi:10.1109/FOCS.2016.56.

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

6 Anthony Bonato and Ehsan Chiniforooshan. Pursuit and evasion from a distance: Algorithms and bounds. In *Proceedings of the Meeting on Analytic Algorithmics and Combinatorics (ANALCO)*, pages 1–10, 2009.

7 Anthony Bonato and Richard J. Nowakowski. *The Game of Cops and Robbers on Graphs*, volume 61. American Mathematical Soc., 2011.

8 Sebastian Brandt, Yuval Emek, Jara Uitto, and Roger Wattenhofer. A tight lower bound for the capture time of the cops and robbers game. In *44th International Colloquium on Automata, Languages, and Programming (ICALP), Warsaw, Poland*, 2017.

9 K. Bringmann. Why walking the dog takes time: Fréchet distance has no strongly subquadratic algorithms unless SETH fails. In *Proceedings 55th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 661–670, 2014.

10 K. Bringmann and M. Künnemann. Quadratic conditional lower bounds for string problems and dynamic time warping. In *Proceedings 56th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 79–97, 2015.

11 Karl Bringmann, Allan Grønlund, and Kasper Green Larsen. A dichotomy for regular expression membership testing. In *Proceedings of the 58th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 307–318, 2017. doi:10.1109/FOCS.2017.36.

12 W. G. Brown. On graphs that do not contain a Thomsen graph. *Canad. Math. Bull.*, 9:281–285, 1966.

13 P. L. Chebyshev. Memoire sur les nombres premiers. *J. Math. Pures Appl.*, 17:366–390, 1852.

14 Nancy E. Clarke and Gary MacGillivray. Characterizations of k-copwin graphs. *Discrete Mathematics*, 312(8):1421–1425, 2012. doi:10.1016/j.disc.2012.01.002.

15 P. Erdős, A. Rényi, and V. T. Sós. On a problem of graph theory. *Studia Sci. Math. Hungar.*, 1:215–235, 1966.

16 Fedor V. Fomin, Petr A. Golovach, Jan Kratochvíl, Nicolas Nisse, and Karol Suchan. Pursuing a fast robber on a graph. *Theoretical Computer Science*, 411(7):1167–1181, 2010.

17 Peter Frankl. Cops and robbers in graphs with large girth and Cayley graphs. *Discrete Applied Mathematics*, 17(3):301–305, 1987. doi:10.1016/0166-218X(87)90033-3.

18 Arthur S. Goldstein and Edward M. Reingold. The complexity of pursuit on a graph. *Theor. Comput. Sci.*, 143(1), 1995.

19 Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *Journal of Computer and System Sciences*, 63(4):512–530, 2001.

20 David S. Johnson. The NP-completeness column: An ongoing guide. *Journal of Algorithms*, 4(4):397–411, 1983.

21 Takumi Kasai, Akeo Adachi, and Shigeki Iwata. Classes of pebble games and complete problems. *SIAM Journal on Computing*, 8(4):574–586, 1979.

22 William B. Kinnersley. Cops and robbers is EXPTIME-complete. *Journal of Combinatorial Theory, Series B*, 111:201–220, 2015.

23 William B. Kinnersley. Bounds on the length of a game of cops and robbers. *CoRR*, abs/1706.08379, 2017. arXiv:1706.08379.

24 Linyuan Lu and Xing Peng. On Meyniel's conjecture of the cop number. *Journal of Graph Theory*, 71(2):192–205, 2012.

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

26 Paweł Prałat. When does a random graph have constant cop number? *Australasian Journal of Combinatorics*, 46:285–296, 2010.

27 Alain Quilliot. *Jeux et Pointes Fixes sur les Graphes.* PhD thesis, Université de Paris VI, 1978.

28 I. Reiman. Über ein Problem von K. Zarankiewicz. *Acta. Math. Acad. Sci. Hungary*, 9:269–273, 1958.

29 L. Roditty and V. Vassilevska Williams. Fast approximation algorithms for the diameter and radius of sparse graphs. In *Proceedings 45th ACM Symposium on Theory of Computing (STOC)*, pages 515–524, 2013.

30 Alex Scott and Benny Sudakov. A bound for the cops and robbers problem. *SIAM Journal on Discrete Mathematics*, 25(3):1438–1442, 2011. doi:10.1137/100812963.

31 Larry J. Stockmeyer and Ashok K. Chandra. Provably difficult combinatorial games. *SIAM Journal on Computing*, 8(2):151–174, 1979.

32 J. Tits. Sur la trialité et certains groupes qui s'en déduisent. *Publ. Math. I.H.E.S.*, 2:14–20, 1959.

33 R. Wenger. Extremal graphs with no C^4s, C^6s, or C^{10}s. *J. Comb. Theory Ser. B*, 52(1):113–116, 1991.