Pursuit and evasion from a distance: algorithms and bounds*

Anthony Bonato† Ehsan Chiniforooshan‡

Abstract
Cops and Robber is a pursuit and evasion game played on graphs that has received much attention. We consider an extension of Cops and Robber, distance k Cops and Robber, where the cops win if they are distance at most k from the robber in G. The cop number of a graph G is the minimum number of cops needed to capture the robber in G. The distance k analogue of the cop number, written $c_k(G)$, equals the minimum number of cops needed to win at a given distance k. We supply a classification result for graphs with bounded $c_k(G)$ values and develop an $O(n^{2s+3})$ algorithm for determining if $c_k(G) \leq s$. In the case $k = 0$, our algorithm is faster than previously known algorithms. Upper and lower bounds are found for $c_k(G)$ in terms of the order of G. We prove that

$$\Omega\left(\left(\frac{n}{k}\right)^{\frac{1}{3}}\right) = c_k(n) = O\left(\frac{n}{\log\left(\frac{2n}{k+1}\right)}\frac{\log(k+2)}{k+1}\right),$$

where $c_k(n)$ is the maximum of $c_k(G)$ over all n-node connected graphs.

1 Introduction and main results

Originating with the work of Nowakowski and Winkler [19], Quilliot [20], and Aigner and Fromme [1] in the 1980's on the game of Cops and Robber, a large and diverse corpus of research has now emerged on pursuit and evasion games on graphs. In pursuit and evasion games, the usual setting is a discrete-time two-person game consisting of an intruder who is loose on the nodes of a graph and trying to evade capture, and a set of searchers whose goal is to capture the robber while minimizing resources. Networks that require a smaller number of searchers may be viewed as more secure than those where many searchers are needed. Variations allow for players to possess only imperfect information, utilize only certain types of movements, allowing the players to move at various speeds, or meet specified conditions to win the game. For example, as is the case in this work, a searcher need not occupy the node of the robber to capture him, but must "see" or "shoot" the robber from some prescribed distance away. For recent surveys on pursuit and evasion games, the reader is directed to [2, 10, 13].

We give a formal description of the game of distance k Cops and Robber, by first recalling how Cops and Robber is played. In Cops and Robber, there are two players, a set of k *cops* (or *searchers*) $\mathcal{C}$, where $k > 0$ is a fixed cardinal, and the *robber* $\mathcal{R}$. The cops begin the game by occupying a set of k nodes of a simple, undirected, finite, connected graph G, and the cops and robber move in *rounds* indexed by nonnegative integers. Each round consists of a cop's move followed by a robber's move. More than one cop is allowed to occupy a node, and the players may *pass*; that is, remain on their current node. A *move* in a given round for a cop or the robber consists of a pass or moving to an adjacent node; each cop may move or pass in a round. The players know each others current locations and can remember all the previous moves; that is, the game is played with *perfect information.* The cops win and the game ends

*Supported by NSERC, MITACS, and the Canada Research Chairs Program.

†Department of Mathematics, Ryerson University, Toronto ON, Canada M5B 2K3; abonato@ryerson.ca

‡Department of Computer Science and Software Engineering, Concordia University, Montréal, QC, Canada H3G 1M8; ehsan@encs.concordia.ca

if at least one of the cops can eventually occupy the same node as the robber; otherwise, $\mathcal{R}$ wins. As placing a cop on each node guarantees that the cops win, we may define the *cop number*, written $c(G)$, which is the minimum of the cardinalities of the set of cops needed to win on G. While this node pursuit game played with one cop was introduced in [19, 20], the cop number was first introduced [1].

In this extended abstract, we study a variation of the game of Cops and Robber in which cops have the ability of catching the robber if he is sufficiently close. More precisely, fix a fixed nonnegative integer parameter k. The game of *distance k Cops and Robber* is played in an analogous as is Cops and Robber, except that the cops win if a cop is within distance at most k from the robber (for simplicity, we identify the players with the nodes they occupy). If $k = 0$, then distance k Cops and Robber reduces to the classical Cops and Robber game.

The minimum value of number of cops which possess a winning strategy in G playing distance k Cops and Robber is denoted by $c_k(G)$. Hence, $c_0(G)$ is just the usual cop number $c(G)$. For example, for the 4-cycle, $c_0(C_4) = 2$, while $c_k(C_4) = 1$ for all $k \geq 1$. Note that for G connected, $c_k(G) = 1$ if $k \geq \text{diam}(G) - 1$, where $\text{diam}(G)$ is the diameter of G. Further, for all $k \geq 1$, $c_k(G) \leq c_{k-1}(G)$. We note that for a given integers $k, m \geq 1$, there are examples of graphs with the property that $c_k(G) = 1$ but $c(G) = m$. For example, if $p \in (0,1)$ is constant, then the random graph $G(n,p)$ asymptotically almost surely (a.a.s.) satisfies $c(G(n,p)) = \Theta(\log n)$ (see [6]), but $c_k(G(n,p)) = 1$ for all $k > 0$ since a.a.s. it has diameter 2. We let $c_k(n)$ denote the maximum of $c_k(G)$ over all n-node connected graphs.

In the case $k = 0$, polynomial-time algorithms were given in [4, 12, 14] for recognizing if G satisfies $c_0(G) \leq s$, where s is a fixed positive integer. In particular, the algorithm of [14] runs in time $O(n^{5s})$, where $n = |V(G)|$. A difficult open problem in graph searching is Meyniel's conjecture (communicated by Frankl [11]), which states that $c_0(G) = O(\sqrt{n})$, where n is the order of G. The best known upper bound for general graphs was given in [8] where it was proved that $c_0(n) = O(\frac{n}{\log n})$ (here $\log n$ is the natural logarithm). Meyniel's conjecture has been essentially verified for $G(n,p)$ random graphs for several cases when p is a function of n; see [5, 6, 7, 18].

We consider both algorithms and bounds for $c_k(G)$. In Section 2, we analyze the complexity of computing $c_k(G)$ for a given graph G. We give a polynomial-time algorithm for determining whether $c_k(G)$ is equal to s, assuming that s is not a part of the input. Our algorithm runs in time $O(n^{2s+3})$ (see Theorem 2.3), and is therefore, the fastest algorithm we are aware of for computing the cop number. For any two integers s and k, Theorem 2.1 gives a classification of the family of graphs with $c_k(G) > s$ using the strong product of graphs. In Sections 3 and 4, we supply upper (see Theorem 3.1) and lower bounds (see Theorem 4.1), respectively, for $c_k(G)$ in terms of the order of G. In particular, we prove that

$$\Omega\left(\left(\frac{n}{k}\right)^{\frac{1}{3}}\right) = c_k(n) = O\left(\frac{n}{\log\left(\frac{2n}{k+1}\right)}\frac{\log(k+2)}{k+1}\right).$$

These bounds generalize known bounds for the cop number, but require new techniques which are of interest in their own right.

All graphs we consider are simple, undirected, finite, connected, and reflexive, unless otherwise stated. The kth *closed neighbourhood of a node x* in G, written ${N_G}^k[x]$, consists of all nodes of distance at most k from x in G, including the node x itself; in the case $k = 1$, we write simply $N[x]$. The kth *closed neighbourhood of a set $X \subseteq V(G)$* is written ${N_G}^k[X]$, and is defined in the analogous way. For $X \subseteq V(G)$, we write $G[X]$ for the subgraph induced by X. For two nodes $x, y \in V(G)$, $d_G(x,y)$ denotes the distance between x and

y in G; we omit the subscript if G is clear from context. A *homomorphism* from G to H is a function $f : V(G) \to V(H)$ such that $xy \in E(G)$ implies that $f(x)f(y) \in E(H)$. A *retraction* f is a homomorphism from G to an induced subgraph H such that $f(x) = x$ for all $x \in V(H)$; the induced subgraph H is called a *retract* of G. For more on homomorphisms and retracts, the reader is directed to [15]. For references on graph theory, the reader is directed to [9, 21]. For a set X and a positive integer s, let X^s denote the sth Cartesian power of X. For an ordered s-tuple T in $V(G)^s$ and an integer $1 \leq i \leq s$, we use T_i to denote the ith element of T.

2 Algorithms for distance k-cop number

We first investigate the complexity of computing $c_k(G)$ for a given graph G. In particular, we show that there is a polynomial-time algorithm that can determine whether $c_k(G) \leq s$ assuming that s is not a part of the input. Our algorithm relies heavily on the following theorem which gives a classification using strong products of the family of graphs with $c_k(G) > s$, for any two integers k and s. Given graphs G and H, their *strong product*, written $G \boxtimes H$, has nodes $V(G) \times V(H)$, with (u_1, u_2) joined to (v_1, v_2) if for $i = 1, 2$, u_i is joined or equal to v_i. We may iterate this product in the obvious way so there are more than two factors. Given a graph G, define the sth *strong power of* G, written $\boxtimes^s G$, to be the strong product of G with itself s times. Using the strong products of graphs for computing the cop number is also implicitly mentioned in [14]; however, their way of using strong products of graphs is different from ours.

THEOREM 2.1. *Suppose that $k, s \geq 0$ are two integers. Then $c_k(G) > s$ if and only if there is a mapping $\psi : V(\boxtimes^s G) \mapsto 2^{V(G)}$ with the following two properties.*

1. *For every $T \in V(\boxtimes^s G)$,*
$$\emptyset \neq \psi(T) \subseteq V(G) \setminus N_G^{k+1}[T].$$

2. *For every $TT' \in E(\boxtimes^s G)$,*
$$\psi(T) \subseteq N[\psi(T')].$$

Proof. For a robber $\mathcal{R}$, we call a sequence $T^{(0)}, r^{(0)}, T^{(1)}, r^{(1)}, \ldots, T^{(t)}, r^{(t)}$ an *$\mathcal{R}$-valid sequence* if $T^{(i)} \in V(\boxtimes^s G)$, $T^{(i)}T^{(i+1)} \in E(\boxtimes^s G)$, $r^{(i)} \in V(G)$, $r^{(i)}r^{(i+1)} \in E(G)$, and $r^{(i)}$s are played according to $\mathcal{R}$'s strategy. In other words, the sequence of moves for $\mathcal{C}$ and $\mathcal{R}$ (alternating between them, with the cops going first) in the first t rounds will be $T^{(0)}, r^{(0)}, T^{(1)}, r^{(1)}, \ldots, T^{(t)}, r^{(t)}$. If $\mathcal{R}$ has a winning strategy, then define $\psi(T)$ for $T \in V(\boxtimes^s G)$ to be the set of all nodes $r \in V(G)$ such that there exist an integer t and an $\mathcal{R}$-valid sequence $T^{(0)}, r^{(0)}, T^{(1)}, r^{(1)}, \ldots, T^{(t)} = T, r^{(t)} = r$. First, to show that $\psi(T) \neq \emptyset$, observe that $\mathcal{C}$ can put the cops at T in round 0. Since $\mathcal{R}$ has a winning strategy, $\mathcal{R}$ must put the robber in some node $r \in V(G)$. Consequently, $r \in \psi(T)$, and thus, $\psi(T) \neq \emptyset$. To show that $\psi(T) \subseteq V(G) \setminus N_G^{k+1}[T]$, assume r is in $\psi(T)$; that is, there is an $\mathcal{R}$-valid sequence $T^{(0)}$, $r^{(0)}$, $T^{(1)}$, $r^{(1)}$, $\ldots$, $T^{(t)} = T$, $r^{(t)} = r$. Then r cannot be in $N_G^{k+1}[T]$; otherwise, $\mathcal{C}$ can capture the robber in round $t + 1$, which contradicts with the fact that $\mathcal{R}$ is playing according to a winning strategy. The first property of the theorem follows.

To prove the second property, let T, T' be an edge in $E(\boxtimes^s G)$ and $r \in \psi(T)$. Then, there exists an $\mathcal{R}$-valid sequence $T^{(0)}$, $r^{(0)}$, $T^{(1)}$, $r^{(1)}$, $\ldots$, $T^{(t)} = T$, $r^{(t)} = r$. Since $TT' \in E(\boxtimes^s G)$, $\mathcal{C}$ can move the cops from T to T' in round $t + 1$. Since $\mathcal{R}$ has a winning strategy, $\mathcal{R}$ must be able to move the robber from r to a node r' that is adjacent or equal to r. Therefore, $r' \in \psi(T')$. Since every node r of $\psi(T)$ has a neighbour $r' \in \psi(T')$, we have $\psi(T) \subseteq N[\psi(T')]$.

Suppose now that a mapping ψ exists with properties 1 and 2. We show that $\mathcal{R}$ has a strategy to avoid capture. Let $T^{(0)} \in V(\boxtimes^s G)$ be the positions of the k cops in round 0; that is, $T_i^{(0)} \in V(G)$ is the position of the ith cop, for all $1 \leq i \leq s$. In round 0, the robber $\mathcal{R}$ moves to an arbitrary node in $\psi(T^{(0)})$. This is possible, because the first property of ψ says that $\psi(T^{(0)}) \neq \emptyset$. In round 0 the cops cannot capture the robber since by the first property of ψ, the nodes of $\psi(T^{(0)})$ have distance at least $k+2$ from any cop in $T^{(0)}$.

We argue that for all $t \geq 0$ the robber can go to $\psi(T^{(t)})$ in round t, where $T^{(t)}$ is the position of s cops in round t. Suppose this claim is true for $t \leq a$. We prove that the claim is true for $a+1$. In each round a cop can move to an adjacent node, so

$$T^{(a)}T^{(a+1)} \in E(\boxtimes^s G).$$

Therefore, by the second property of ψ, $\psi(T^{(a+1)}) \subseteq N[\psi(T^{(a)})]$. Hence, the robber at $\psi(T^{(a)})$ can move to a node in $\psi(T^{(a+1)})$ in round $a+1$ and avoid capture. □

We now consider a polynomial-time algorithm for determining whether $c_k(G) \leq s$.

Algorithm 1 CHECK-DISTANCE-COP-NUMBER-S

Require: $G = (V, E)$, $s \geq 0$

1: initialize $\psi(T)$ to $V(G) \setminus N_G^{k+1}[T]$, for all $T \in V(\boxtimes^s G)$
2: **repeat**
3:    **for all** $TT' \in E(\boxtimes^s G)$ **do**
4:       $\psi(T) \leftarrow \psi(T) \cap N_G[\psi(T')]$
5:       $\psi(T') \leftarrow \psi(T') \cap N_G[\psi(T)]$
6:    **end for**
7: **until** the value of ψ is unchanged
8: **if** there exists $T \in V(\boxtimes^s G)$ such that $\psi(T) = \emptyset$ **then**
9:    **return** $c_k(G) \leq s$
10: **else**
11:    **return** $c_k(G) > s$
12: **end if**

THEOREM 2.2. *Algorithm 1 runs in time $O(n^{3s+3})$.*

Proof. We may determine if there exists a mapping ψ with properties stated in Theorem 2.1 using Algorithm 1. It is clear that if the algorithm terminates, it will answer correctly; either it finds a ψ with properties stated in Theorem 2.1, or no such ψ exists because nothing from $\psi(T)$ will be removed unless it is necessary. In other words, for any mapping ψ' with properties stated in Theorem 2.1 we will have $\psi'(T) \subseteq \psi(T)$, for all $T \in V(\boxtimes^s G)$, where ψ is the mapping found by Algorithm 1. Hence, if $\psi(T) = \emptyset$ for some T, there is no mapping with the stated properties. The running-time of Algorithm 1 is at most $O(n^{3s+3})$, since the repeat loop in lines 2–7 iterates at most $O(n^{s+1})$ times. This is because at each iteration, except the last one, the cardinality of $\psi(T)$ will be decreased for at least one T. □

We may implement Algorithm 1 in a more efficient way to reduce the running time. Algorithm 2 determines if there exists a mapping ψ with properties stated in Theorem 2.1 in time $O(n^{2s+3})$. We prove this claim in Theorem 2.3. Note that Algorithm 2 is not only more general than previously known algorithms for answering $c_0(G) \leq s$ (since it can determine $c_k(G) \leq s$ for any k), it is also faster. The algorithm in [14] runs in time $O(n^{5s})$.

THEOREM 2.3. *Algorithm 2 runs in time $O(n^{2s+3})$.*

Proof. There are some details that are left out in the algorithm, such as computing set intersections and neighbourhoods. Set intersection and difference can be done in time $O(n)$ if the sets are of cardinality at most n. We assume that the algorithm computes $N_G[v]$ and $N_G^{k+1}[v]$ for each vertex $v \in V(G)$ in a one-time preprocessing. This will not affect the total running time of the algorithm. In this way, computing

Algorithm 2 CHECK-DISTANCE-COP-NUMBER-S

Require: $G = (V, E)$, $s \geq 0$

1: initialize $\psi(T)$ to $V(G) \setminus N_G^{k+1}[T]$, for all $T \in V(\boxtimes^s G)$
2: initialize the queue Q to contain $V(\boxtimes^s G)$
3: **while** Q is not empty **do**
4:    pop T from the head of Q
5:    **for all** neighbours T' of T **do**
6:       $\psi(T') \leftarrow \psi(T') \cap N_G[\psi(T)]$
7:       **if** $\psi(T')$ is changed **then**
8:          add T' to the end of Q
9:       **end if**
10:    **end for**
11: **end while**
12: **if** there exists $T \in V(\boxtimes^s G)$ such that $\psi(T) = \emptyset$ **then**
13:    **return** $c_k(G) \leq s$
14: **else**
15:    **return** $c_k(G) > s$
16: **end if**

$N_G[T]$ and $N_G^{k+1}[T]$ can be done in $O(n^2)$. As a one-time preprocessing, the algorithm keeps a list of all neighbors of T in $\boxtimes^s G$, for each $T \in V(\boxtimes^s G)$. This will take at most time $O(n^{2s+1})$.

We now analyze the running-time of Algorithm 2: lines 1–2, and 12–16 take time at most $O(n^{s+2})$. Lines 6–9 take $O(n^2)$, and thus, the for loop in lines 5–10 takes time $O(n^{s+2})$. Line 4 can be done in constant time. Hence, the total running-time of the algorithm is $O(n^{s+2}x + n^{2s+1})$, where x is the maximum number of iterations of the while loop. Note that after each iteration of the while loop, the value of $|Q| + \sum_{T \in V(\boxtimes^s G)} \psi(T)$ will be decreased by at least one. Consequently, x is at most $O(n^{s+1})$ and the theorem follows. □

3 Upper bounds for $c_k(n)$

Our main result in this section is the following upper bound on $c_k(n)$.

THEOREM 3.1. *For integers $n > 0$ and $k \geq 0$,*

$$c_k(n) = O\left(\frac{n}{\log\left(\frac{2n}{k+1}\right)} \frac{\log(k+2)}{k+1}\right).$$

From Theorem 3.1,

$$c_0(n) = O\left(\frac{n}{\log n}\right),$$

which is the best known upper-bound for $c_0(n)$ as given in [8].

Before we give the proof of Theorem 3.1, we consider various lemmas. Fix a a positive integer. We let $N_G^i[T]$ denote $N_G^i[\{T_j : 1 \leq j \leq a\}]$, for any $T \in V(\boxtimes^a G)$. A homomorphism φ from G to $\boxtimes^a H$, where H is an induced subgraph of G, is called an *a-guarding function* from G to H if for all $x \in V(H)$,

$$x \in \bigcap_{y \in N_G[x]} N_H[\varphi(y)].$$

Note that $\varphi(x)$ corresponds to an a tuple of nodes of G. Moreover, a subgraph H of G is called *a-guardable* if there is an a-guarding function from G to H.

We note that an induced subgraph H of G is 1-guardable if and only if it is a retract (recall that all the graphs in this paper are assumed to be reflexive). To see this, suppose that φ is a retraction from G to H. Since φ is a homomorphism, $\varphi(x)$ is a neighbour of $\varphi(y)$ if y is a neighbour of x. Therefore,

$$\varphi(x) \in \bigcap_{y \in N_G[x]} N_H[\varphi(y)].$$

Since φ is a retraction, we have that $x = \varphi(x)$ for all $x \in V(H)$, and hence,

$$x \in \bigcap_{y \in N_G[x]} N_H[\varphi(y)]$$

for all $x \in V(H)$. Therefore, H is 1-guardable. Conversely, suppose that φ is a 1-guardable function from G to H. Then φ', defined below, is a retraction from G to H:

$$\varphi'(v) = \begin{cases} v & v \in V(H), \\ \varphi(v) & v \notin V(H). \end{cases}$$

We may therefore view a-guarding functions as generalizations of retractions. The proof of the following theorem is immediate.

Lemma 3.1. *Suppose φ is an a-guarding function from G to H, $x \in V(H)$, and $y \in V(G)$ is a node of distance $k \geq 1$ from x. Then there is at least one node in $\varphi(y)$ whose distance from x in H is at most k; that is, $x \in N_H^k[\varphi(y)]$.*

For any integer $k \geq 0$ and any a-guardable subgraph H of G, define the integer

$$\Lambda(k, G, H) = c_k\left(G[V(G) \setminus N_G^{\lfloor \frac{k}{2} \rfloor}[V(H)]]\right).$$

Lemma 3.2. *For any integer $k \geq 0$ and any a-guardable subgraph H of G,*

$$c_k(G) \quad \leq \quad a + \max\{\Lambda(k, G, H), c(H) - 1\}.$$

Proof. Let φ be an a-guarding function from G to H. The strategy for $\mathcal{C}$ is the following: using $c(H) + a - 1$ cops, $\mathcal{C}$ can eventually move, say at round t_0, a cops to the image of the robber in H; that is, $\varphi(r)$, where r is the position of the robber. This is possible because $\mathcal{C}$ can chase $\varphi_1(r)$ in H using $c(H)$ cops and eventually put the first cop at $\varphi_1(r)$. Then $\mathcal{C}$ keeps one cop at $\varphi_1(r)$ and starts to chase $\varphi_2(r)$ using $c(H)$ unused cops, and so on. The above-mentioned a cops will remain at $\varphi(r)$ at all the times $t \geq t_0$, unless they can capture the robber in one move, in which case they do so instead of going to $\varphi(r)$.

Now, suppose the robber moves to a node $r \in N_G^{\lfloor \frac{k}{2} \rfloor}[V(H)]$ at round $t > t_0$. Then there is a node $x \in V(H)$ of distance $\ell \leq \lfloor \frac{k}{2} \rfloor$ from r. If $\ell \geq 1$, by Lemma 3.1, x is in $N_H^{\ell}[\varphi(r)]$, and thus, $r \in N_G^{2\ell}[\varphi(r)] \subseteq N_G^k[\varphi(r)]$. Therefore, since the distance of r and $\varphi(r)$ is at most k, and there are cops at all the nodes of $\varphi(r)$ at round $t+1$, the robber is captured at round $t+1$.

In the case that $k = 0$, that is, $x = r$, let r' be the position of the robber at round $t-1$. We know that $r' \in N_G[r] = N_G[x]$. Then by the definition of a-guarding function, $r = x \in N_H[\varphi(r')]$ and since there are cops in all the nodes of $\varphi(r')$ in round t, one cop in $\varphi(r')$ can move to r and capture the robber at round $t+1$.

The above argument shows that the robber cannot move to any node in $N_G^{\lfloor \frac{k}{2} \rfloor}[V(H)]$ after round t_0. But then $\mathcal{C}$ can capture the robber in the induced subgraph $G[V(G) \setminus N_G^{\lfloor \frac{k}{2} \rfloor}[V(H)]]$ using $\max\{\Lambda(k, G, H), c(H) - 1\}$-many cops. □

Lemma 3.2 says that we can remove the nodes of $\lfloor \frac{k}{2} \rfloor$-neighbourhood of H from G at the cost of at most a cops, that is, $N_G^{\lfloor \frac{k}{2} \rfloor}[H]$ can be "guarded" by a cops. For a given a and k, how large can the closed $\lfloor \frac{k}{2} \rfloor$-neighbourhood of an a-guardable subgraph be? The following lemma answers this question and was implicit in [8].

Lemma 3.3. *If P is a shortest path in G, then a subgraph H containing P such that $V(H) \subseteq N_G[P]$ is 5-guardable.*

Proof. Let the nodes of P be $p_1, p_2, \ldots, p_\ell$. The homomorphism φ, defined below, is a 5-guarding function from G to H: for all $1 \leq i \leq 5$,

$$\varphi_i(v) = \begin{cases} p_0 & d(v, p_1) + i < 4, \\ p_\ell & d(v, p_1) + i - 3 > \ell, \\ p_{d(v,p_1)+i-3} & \text{otherwise.} \end{cases}$$ □

We use Lemma 3.3 together with the following lemma to obtain 5-guardable subgraphs with large neighbourhoods.

LEMMA 3.4. *For any two integers $n, d \geq 1$ and any rooted n-node tree T, T has a root-to-leaf path P such that*

$$\left|N_T^d[P]\right| \;\geq\; \frac{d\log(1+\frac{n}{d})}{1+\log d}.$$

Proof. Let $\tau(n,d)$ be the largest number such that any rooted n-node tree T has a root-to-leaf path P such that $|N_T^d[P]| \geq \tau(n,d)$. We use induction on n to prove that $\tau(n,d) \geq \frac{d}{1+\log d}\log(1+\frac{n}{d})$. As for the base case, it is clear that for all $1 \leq n \leq 2d$, $\tau(n,d) = n \geq \frac{d}{1+\log d}\log(1+\frac{n}{d})$.

We assume that the hypothesis is true for all integers up to $n \geq 2d$ and we prove that $\tau(n+1) \geq \frac{d}{1+\log d}\log(1+\frac{n+1}{d})$. So, let T be an $n+1$-node tree in which all root-to-leaf paths P have $|N_T^d[P]| \leq \tau(n+1,d)$, r be the root of T, B_i be the set of nodes of distance at most i from r, and $b_i = |B_i|$. We can assume that $b_d - b_{d-1} > 0$, otherwise, if $b_d = b_{d-1}$, all the nodes of T are at distance at most $d-1$ of r, and thus, $\tau(n+1,d) \geq |N_T^d[r]| = n+1 \geq \frac{d}{1+\log d}\log(1+\frac{n+1}{d})$. Since any path of length $d-1$ has d nodes, $b_{d-1} \geq d$. Let $v \in B_d \setminus B_{d-1}$ be the node that maximized the number of nodes in T_v, the subtree of T rooted at v. Clearly, $|V(T_v)| \geq \frac{n+1-b_{d-1}}{b_d-b_{d-1}}$. Therefore, there is a path P_v in T_v from v to a leaf such that $|N_{T_v}^d[P_v]| \geq \tau(\frac{n+1-b_{d-1}}{b_d-b_{d-1}}, d)$. Let $P_{r,v}$ denote the path from r to v in T from which v is removed. By joining $P_{r,v}$ and P_v we obtain a root-to-leaf path P in T, and have that

$$\begin{aligned}
\tau(n+1,b) &\geq \left|N_T^d[P]\right| \\
&\geq \tau\left(\frac{n+1-b_{d-1}}{b_d-b_{d-1}}, d\right) + b_d - 1 \\
&\geq \tau\left(\frac{n+1-d}{b_d-d}, d\right) + b_d - 1 \\
&\geq \frac{d\log\left(1-\frac{1}{b_d-d}+\frac{n+1}{d(b_d-d)}\right)}{1+\log d} + b_d - 1 \\
&= \frac{d\log\left(\left(1-\frac{1}{b_d-d}\right)(2d)^{\frac{b_d-1}{d}}+\frac{(2d)^{\frac{b_d-1}{d}}}{b_d-d}\,\frac{n+1}{d}\right)}{1+\log d} \\
&\geq \frac{d\log\left(1+\frac{n+1}{d}\right)}{1+\log d}. \qquad \square
\end{aligned}$$

The lower bound of $\frac{d\log(1+\frac{n}{d})}{1+\log d}$ is not necessarily tight; however, it cannot be larger than $2d\log(1+\frac{n}{d})$, as it can been verified in a complete binary tree in which all the edges are subdivided $d-1$ times; see Figure 1.

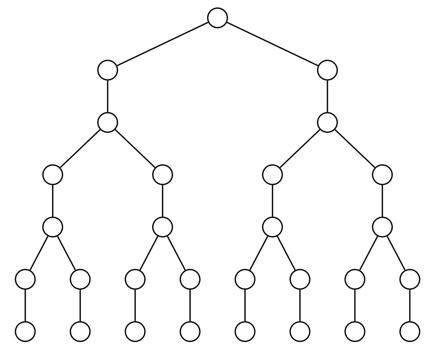

Figure 1: A rooted tree showing that $\tau(n,d) \leq 2d\log(1+\frac{n}{d})$, where $n = 29$ and $d = 2$.

With Lemma 3.4 and Lemma 3.3 we now may prove Theorem 3.1.

Proof of Theorem 3.1. Let G be an n-node connected graph and T be a rooted spanning BFS tree of G (see [17] for the definition of BFS trees). By Lemma 3.4, T has a root-to-leaf path P, such that $|N_T^d[P]| \geq \frac{d\log(1+\frac{n}{d})}{1+\log d}$, where $d = 1 + \left\lfloor\frac{k}{2}\right\rfloor$. Since T is a BFS tree, P is a shortest path in G. Let T' be any spanning tree of $G[N_G[P]]$ that contains P. Now T' is 5-guardable, due to Lemma 3.3. Since $c(T') = 1$, we can use Lemma 3.2 to obtain that

$$\begin{aligned}
c_k(n) &\leq c_k\left(G[V(G)\setminus N_G^{1+\left\lfloor\frac{k}{2}\right\rfloor}[P]]\right) + 5 \\
&\leq c_k\left(n - \frac{d\log\left(1+\frac{n}{d}\right)}{1+\log d}\right) + 5.
\end{aligned}$$

Therefore,

$$c_k(n) \;=\; O\left(\frac{n\left(1+\log d\right)}{d\log\left(1+\frac{n}{d}\right)}\right)$$

$$= O\left(\frac{n}{\log\left(\frac{2n}{k+1}\right)}\frac{\log(k+2)}{k+1}\right). \qquad \square$$

4 Lower bounds for $c_k(n)$

By considering graphs arising from projective planes it was noted in [5, 18] that

$$c_0(n) = \Omega(\sqrt{n}).$$

However, by Theorem 3.1, it is clear that for large values of k, $c_k(n)$ may be much less than $\Omega(\sqrt{n})$. In this section, we establish a lower bound for $c_k(n)$ in terms of n and k. We note that few lower bounds are known for the cop number in terms of familiar graph parameters. One such lower bound was found by Frankl, who gave lower bounds on $c(G)$ in the case of large girth graphs; see [11].

Given a graph G and a positive integer ℓ, form G^ℓ by replacing each edge of G by a path with ℓ edges. For example, K_4^2 is illustrated in Figure 4. For simplicity, we identify nodes of G with corresponding nodes in G^ℓ; in particular, $V(G) \subseteq V(G^\ell)$. Nodes of G^ℓ that are not in G are called *internal nodes*.

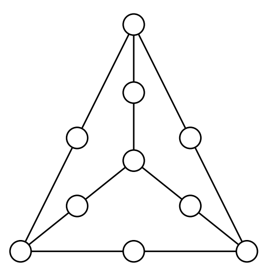

Figure 2: The graph K_4^2.

LEMMA 4.1. *For any graph G and any integer $k \geq 0$,*

$$c_k(G^{(2k+1)}) \geq c(G).$$

Lemma 4.1 sets up a relationship between the cop number $c(G)$ and $c_k(G)$. We note that the inequality in the lemma is not tight in general: for example, $c_1((K_3)^3) = 2$ while $c(K_3) = 1$. Further, the lemma may be generalized so that edges of G are replaced by isometric subgraphs with fixed diameter. We will explore this generalization in the full version of this extended abstract.

Proof of Lemma 4.1. Let $c = c(G) - 1$. The robber $\mathcal{R}$ has a winning strategy in Cops and Robber played on G if there are only c cops. We will show that $\mathcal{R}$ has a winning strategy in distance k Cops and Robber played on $G^{(2k+1)}$ if there are only c cops.

For each internal node $x \in V(G^{(2k+1)})$ there is exactly one node in $V(G)$ whose distance from x is at most k; name this node x_k. Define a function f from the nodes of $G^{(2k+1)}$ to nodes of G that is the identity on $V(G)$, so that if x is internal node, then $f(x) = x_k$. The robber $\mathcal{R}$ simulates the winning strategy for Cops and Robber played on G in distance k Cops and Robber played on $G^{(2k+1)}$ by using the function f, and will play in a way that the robber will always be in $V(G)$ in rounds

$$2k, 4k+1, \ldots, 2ik+i-1, \ldots$$

for all $i \geq 1$.

In round 0, $\mathcal{C}$ puts c cops in $v_1, v_2, \ldots, v_c$. In round 0, $\mathcal{R}$ assumes that the cops are at $f(v_1), f(v_2), \ldots, f(v_c)$ and puts the robber in a node $r \in V(G)$ pretending that the game is being played in G. Since the robber would not be captured in G, neither of $f(v_i)$'s are adjacent to r in G, and hence, v_i's are of distance at least $3k+2$ from r in $G^{(2k+1)}$. Therefore, the cops cannot capture the robber in rounds $0 \leq t \leq 2k+1$, if the robber stays at r in rounds $0 \leq t \leq 2k$.

Let $v_1', v_2', \ldots, v_c'$ be the positions of cops in round $2k+1$. In $2k+1$ rounds, for each $1 \leq i \leq c$ we will have either $f(v_i) = f(v_i')$ or $f(v_i)$ is adjacent to $f(v_i')$ in G. Thus, $\mathcal{R}$ can assume that $\mathcal{C}$ has moved the cops from $f(v_1), f(v_2), \ldots, f(v_c)$ to $f(v_1'), f(v_2'), \ldots, f(v_c')$ in G in one round. Let r' be the node to which

$\mathcal{C}$ would move the robber if the game was being played in G. The strategy of $\mathcal{R}$ in $G^{(2k+1)}$ is to move the robber from r to r' in the next $2k+1$ rounds. It is easy to verify that the cops cannot capture the robber in the next $2k+1$ rounds and, in round $4k+2$, the robber can decide the next $2k+1$ rounds. The rest follows by induction. □

Lemma 4.1 gives us a tool for transferring lower bounds on $c(n)$ to lower bounds on $c_k(n)$.

THEOREM 4.1. *For all $k \geq 0$ and $n \geq 1$ integers, we have that*

$$c_k(n) = \Omega\left(\left(\frac{n}{k}\right)^{\frac{1}{3}}\right).$$

Proof. Let $G(q)$ be the *incidence graph of a projective plane of order* q; that is, the bipartite graph with nodes the points and lines of the geometry, and with two nodes representing a point and a line being adjacent if the point is on the line. If the geometry has q points, then the $G(q)$ has $2(q^2+q+1)$ nodes. The cop number of $G(q)$ is at least $q+1$. This fact was noted in [5], but we include a proof for completeness. Note that the girth of $G(q)$ is at least 5 and that $G(q)$ is $(q+1)$-regular; now apply Theorem 3 of [1]. Therefore, $c_k(G^{(2k+1)}) \geq q+1$, by Lemma 4.1. The proof follows since

$$|V(G^{(2k+1)})| = 2(q^2+q+1)(kq+k+1). \quad \square$$

5 Conclusions and further work

We introduced a new variant of the Cops and Robber game where the cops win if they are within distance k of the robber, and subsequently defined a new graph parameter $c_k(G)$. For an integer $s \geq 1$, we supplied a polynomial-time algorithm recognizing when $c_k(G) \leq s$. Upper and lower bounds for $c_k(G)$ were supplied extending known results on the cop number.

As the parameter $c_k(G)$ is new, there are many open problems surrounding its properties. In all algorithms that we are aware of for determining if $c_k(G) \leq s$, s appears in the exponent of n. Usually, it is preferred to find algorithms with running times of the form $O(f(s)n^\alpha)$, where $f(s)$ is any function depending only on s (independent of n) and α is a constant, independent of n and s. Such algorithms are called *fixed-parameter algorithms* with parameter s. Thus, the following question is open: is there a fixed-parameter algorithm with parameter s for answering $c_k(G) \leq s$? A modified version of Algorithm 1 will be presented in the full version of this extended abstract which computes $c_k(G)$ in time $O(\text{poly}(n)8^n)$. It would be interesting to know what is the fastest (likely exponential-time algorithm) to compute $c_k(G)$, when $c_k(G)$ is not bounded by a constant s. That is, what is the smallest value of α such that there is an algorithm for computing $c_k(G)$ that runs in time $O(\text{poly}(n)\alpha^n)$?

In Theorem 3.1 we proved that

$$m = O\left(\frac{n}{\log\left(\frac{2n}{k+1}\right)} \frac{\log(k+2)}{k+1}\right)$$

cops are enough to capture the robber in an n-node graph. One question is to determine the complexity of the problem where the cops must capture the robber using m cops. It is not hard to see that all the steps described in the proof of Theorem 3.1 can be done in polynomial-time; in other words, there is a polynomial-time algorithm that can partition any graph G into m-many 5-guardable subgraphs. We skip the detailed analysis of the running-time of such an algorithm.

There is a gap between the upper bound in Theorem 3.1 and the lower bound in Theorem 4.1. Hence, it is open to find tighter upper bounds or lower bounds for $c_k(n)$. One method to improve the lower bound is to find sparse n-vertex graphs with high cop-number and use them in conjunction with Lemma 4.1. Incidence graphs of a projective planes have cop-number $O(\sqrt{n})$ but are dense with $O(n^{1.5})$-many edges.

Cop-win graphs, where one cop wins, were structurally characterized in [19, 20]. The cop-win graphs are exactly those graphs which are *dismantlable*: there exists a linear ordering $(x_j : 1 \leq j \leq n)$ of the nodes so that for all $2 \leq j \leq n$, there is a $i < j$ such that $N[x_j] \subseteq N[x_i]$. For instance, chordal and bridged graph are cop-win; see [3]. No analogous structural characterization of graphs G satisfying $c_k(G) = 1$, where $k > 1$ is a fixed integer, is known.

References

[1] M. Aigner, M. Fromme, A game of cops and robbers, *Discrete Applied Mathematics* **8** (1984) 1-12.

[2] B. Alspach, Sweeping and searching in graphs: a brief survey, *Matematiche* **59** (2006) 5-37.

[3] R.P. Anstee, M. Farber, On bridged graphs and cop-win graphs, *Journal of Combinatorial Theory, Series B* **44** (1988) 22-28.

[4] A. Berarducci, B. Intrigila, On the cop number of a graph, *Advances in Applied Mathematics* **14** (1993) 389-403.

[5] B. Bollobás, G. Kun, I. Leader, Cops and robbers in a random graph, submitted.

[6] A. Bonato, G. Hahn, C. Wang, The cop density of a graph, *Contributions to Discrete Mathematics* **2** (2007) 133-144.

[7] A. Bonato, P. Prałat, C. Wang, Network security in models of complex networks, submitted.

[8] E. Chiniforooshan, A better bound for the cop number of general graphs, *Journal of Graph Theory* **58** (2008) 45-48.

[9] R. Diestel, *Graph theory*, Springer-Verlag, New York, 2000.

[10] F.V. Fomin, D. Thilikos, An annotated bibliography on guaranteed graph searching, *Theoretical Computer Science* **399** (2008) 236-245.

[11] P. Frankl, Cops and robbers in graphs with large girth and Cayley graphs, *Discrete Applied Mathematics* **17** (1987) 301–305.

[12] A.S. Goldstein, E.M. Reingold, The complexity of pursuit on a graph, *Theoretical Computer Science* **143** (1995) 93–112.

[13] G. Hahn, Cops, robbers and graphs, *Tatra Mountain Mathematical Publications* **36** (2007) 163-176.

[14] G. Hahn, G. MacGillivray, A characterization of k-cop-win graphs and digraphs, *Discrete Mathematics* **306** (2006) 2492-2497.

[15] P. Hell, J. Nešetřil, *Graphs and homomorphisms*, Oxford University Press, Oxford, 2004.

[16] W. Imrich, S. Klavzar, *Product graphs-Structure and recognition*, Wiley-Interscience series in discrete mathematics and optimization. Wiley-Interscience, New York, 2000.

[17] D.E. Knuth, *The art of computer programming*, Vol 1, Addison-Wesley, 1997.

[18] T. Łuczak, P. Prałat, Chasing robbers on random graphs: zigzag theorem, submitted.

[19] R. Nowakowski, P. Winkler, Vertex-to-vertex pursuit in a graph, *Discrete Mathematics* **43** (1983) 235-239.

[20] A. Quilliot, Jeux et pointes fixes sur les graphes, Ph.D Dissertation, Université de Paris VI, 1978.

[21] D.B. West, *Introduction to Graph Theory, 2nd edition*, Prentice Hall, 2001.