COPS AND ROBBERS FROM A DISTANCE

ANTHONY BONATO, EHSAN CHINIFOROOSHAN, AND PAWEŁ PRAŁAT

Abstract. Cops and Robbers is a pursuit and evasion game played on graphs that has received much attention. We consider an extension of Cops and Robbers, distance k Cops and Robbers, where the cops win if at least one of them is of 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 study the parameter c_k from algorithmic, structural, and probabilistic perspectives. We supply a classification result for graphs with bounded $c_k(G)$ values and develop an $O(n^{2s+3})$ algorithm for determining if $c_k(G) \leq s$ for s fixed. We prove that if s is not fixed, then computing $c_k(G)$ is NP-hard. Upper and lower bounds are found for $c_k(G)$ in terms of the order of G. We prove that

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

where $c_k(n)$ is the maximum of $c_k(G)$ over all n-vertex connected graphs. The parameter $c_k(G)$ is investigated asymptotically in random graphs $G(n,p)$ for a wide range of $p = p(n)$. For each $k \geq 0$, it is shown that $c_k(G)$ as a function of the average degree $d(n) = pn$ forms an intriguing zigzag shape.

1. Introduction and main results

Originating with the work of Nowakowski and Winkler [24], Quilliot [25], and Aigner and Fromme [1] in the 1980's on the game of Cops and Robbers, 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 vertices 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. See [12] for a survey of such variations. For example, as is the case in this work, a searcher need not occupy the vertex of the robber to capture him, but must "see" or "shoot" the robber from some prescribed distance away. For analogies from computer gaming, classic Cops and Robbers is akin to a moving-target game where the intruder must be touched to lose (such as Pac-Man),

1991 *Mathematics Subject Classification.* 05C75, 05C85, 05C80.

Key words and phrases. graphs, cop number, polynomial time algorithm, strong product, NP-hard, random graph.

The authors gratefully acknowledge support from NSERC and MITACS. The present article is the full version of an article which appeared in the Proceedings of ANALCO'09.

while the scenarios we consider compare with first-person shooter games where weapons hit targets at some prescribed distance. For recent surveys on pursuit and evasion games, the reader is directed to [2, 12, 15].

We give a formal description of the game of distance k Cops and Robbers, by first recalling how Cops and Robbers is played. In Cops and Robbers, there are two players, a set of s *cops* (or *searchers*) $\mathcal{C}$, where $s > 0$ is a fixed integer, and the *robber* $\mathcal{R}$. The cops begin the game by occupying a set of s vertices of an undirected, and finite graph G. We take G to be *reflexive*: there are loops on each vertex. While the game may be played on a disconnected graph, without loss of generality, assume that G is connected (since the game is played independently on each component and the number of cops required is the sum over all components). The cops and robber move in *rounds* indexed by nonnegative integers. Each round consists movements by one or more cops, followed by a move by the robber. More than one cop is allowed to occupy a vertex, and the players may *pass*; that is, remain on their current vertices. A *move* in a given round for a cop or the robber consists of a pass or moving to an adjacent vertex; each cop may move or pass in a round. The players know each other's current locations; that is, the game is played with *perfect information*. The cops win and the game ends if at least one of the cops can eventually occupy the same vertex as the robber; otherwise, $\mathcal{R}$ wins. Note that if s cops win the game so that in round 0 they occupy a set of vertices S, then they may win by occupying any set of vertices in round 0 (simply move the cops to the vertices of S, and then play as if starting the game at S). As placing a cop on each vertex guarantees that the cops win, we may define the *cop number*, written $c(G)$, which is the minimum cardinality of the set of cops needed to win on G. While this vertex pursuit game played with one cop was introduced in [24, 25], the cop number was first introduced in [1].

We study a variation of the game of Cops and Robbers in which cops have the ability of catching the robber if he is sufficiently close. More precisely, fix a nonnegative integer parameter k. The game of *distance k Cops and Robbers* is played in a way analogous to Cops and Robbers, 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 vertices they occupy). If $k = 0$, then distance k Cops and Robbers reduces to the classical Cops and Robbers game.

The minimum number of cops which possess a winning strategy in G playing distance k Cops and Robbers 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 $k \geq \mathrm{diam}(G) - 1$, where $\mathrm{diam}(G)$ is the diameter of G. Further, for all $k \geq 1$, $c_k(G) \leq c_{k-1}(G)$.

We observe that for given integers $k, m \geq 1$, there are examples of graphs with the property that $c_k(G) = 1$ but $c(G) = m$. To see this, we consider random graphs. The *random graph* $G(n,p)$ consists of the probability space $(\Omega, \mathcal{F}, \mathbb{P})$, where Ω is the set of all graphs with vertex set $[n] = \{1, 2, \dots, n\}$, $\mathcal{F}$ is the family of all subsets of Ω, and for every $G \in \Omega$

$$\mathbb{P}(G) = p^{|E(G)|}(1-p)^{\binom{n}{2} - |E(G)|}.$$

This space may be viewed as $\binom{n}{2}$ independent coin flips, one for each pair of vertices, where the probability of success (that is, drawing an edge) is equal to p. Note that $p = p(n)$ can tend to zero with n. We say that an event holds *asymptotically almost surely* (a.a.s.) if it holds with probability tending to 1 as $n \to \infty$. Now, if $p \in (0, 1)$ is constant, then the random graph $G(n,p)$ a.a.s. satisfies $c(G(n,p)) = \Theta(\log n)$ (see [7]), but a.a.s. $c_k(G(n,p)) = 1$ for all $k > 0$ since a.a.s. it has diameter 2.

In the case $k = 0$, polynomial-time algorithms were given in [4, 14, 16] for recognizing if G satisfies $c_0(G) \le s$, where s is a fixed positive integer. In particular, it is implicit in the work of [16] that their algorithm runs in time $O(n^{2s+3})$, where $n = |V(G)|$.

A difficult open problem in graph searching is Meyniel's conjecture (communicated by Frankl [13]), which states that $c_0(G) = O(\sqrt{n})$. Up until recently, the best known upper bound for general graphs was given in [9] where it was proved that $c_0(n) = O(\frac{n}{\log n})$. Recent work of from [22] proved using the probabilistic method that $c_0(n) = n2^{-(1-o(1))\sqrt{\log n}}$. Meyniel's conjecture has been essentially verified for $G(n,p)$ random graphs for several cases when p is a function of n; see [6, 7, 8, 23].

We study the parameter c_k from algorithmic, structural, and probabilistic perspectives. In particular, we consider both algorithms and bounds for $c_k(G)$, as well as the game played on $G(n,p)$. 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 fixed. Our algorithm runs in time $O(n^{2s+3})$ (see Theorem 3), regardless of the value of k. For any two integers s and k, Theorem 1 gives a classification of the family of graphs with $c_k(G) > s$ using the strong product of graphs. Despite Theorem 3, we prove in Corollary 10 that for any integer $k \ge 0$ there is no polynomial-time algorithm to compute $c_k(G)$, unless P=NP.

In Sections 3 and 4, we supply upper and lower bounds for $c_k(G)$ in terms of the order of G; see Theorems 4 and 11, respectively. We let $c_k(n)$ denote the maximum of $c_k(G)$ over all n-vertex connected graphs. It is shown that

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

These bounds generalize known bounds for the cop number, but require new techniques which are of interest in their own right. In Theorem 12, we present asymptotic results for $c_k(G(n,p))$, where $p = p(n)$. In particular, for each $k \ge 0$, the graph of the function $c_k(G(n,p))$ follows a characteristic zigzag shape (see Figure 3). Theorem 12 and the results of Section 5 generalize the results of [23] which considered the case $k = 0$.

All graphs we consider are undirected, finite, connected, and reflexive (that is, all vertices contain one loop), unless otherwise stated. The kth *closed neighbourhood of a vertex* x in G, written ${N_G}^k[x]$, consists of all vertices of distance at most k from x in G, including the vertex x itself; in the case $k = 1$, we write simply $N_G[x]$. 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 vertices $x, y \in V(G)$, $d_G(x,y)$ 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 [17]. For references on graph theory, the reader is directed to [10, 27]. For background on random graphs see [5, 19]. 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. The set of all subsets of a set X is denoted by 2^X.

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 fixed (that is, not a function of $|V(G)|$ or k). 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 vertices $V(G) \times V(H)$, with (u_1, u_2) adjacent to (v_1, v_2) if for each $i = 1, 2$, u_i is adjacent 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. See [18] for additional background on strong products of graphs. Using the strong products of graphs for computing the cop number is also implicitly mentioned in [16]; however, their use of strong products is different from ours.

Theorem 1. *Suppose that $k \geq 0$, and $s \geq 1$ are integers. Then $c_k(G) > s$ if and only if there is a mapping $\psi : V(\boxtimes^s G) \to 2^{V(G)}$ with the following 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 $TT' \in E(\boxtimes^s G)$,*
$$\psi(T) \subseteq N_G[\psi(T')].$$

Proof. Let s cops play on G. If $\mathcal{R}$ has a winning strategy, then define $\psi(T)$ for $T \in V(\boxtimes^s G)$ to be the set of all vertices $r \in V(G)$ such that if the cops start from the initial position T, then robber can start from r and win the game. Since $\mathcal{R}$ has a winning strategy, $\psi(T)$ is non-empty for every $T \in V(\boxtimes^s G)$. To show that $\psi(T) \subseteq V(G) \setminus N_G^{k+1}[T]$, assume r is in $\psi(T)$. Then r cannot be in $N_G^{k+1}[T]$; otherwise, $\mathcal{C}$ can capture the robber, which contradicts the fact that $\mathcal{R}$ can win the game starting from this configuration.

To prove the second property, let TT' be an edge in $E(\boxtimes^s G)$ and $r \in \psi(T)$. Then, the robber can win if the cops are on T and the robber is on 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 vertex r' that is adjacent or equal to r. Therefore, $r' \in \psi(T')$. Since every vertex r of $\psi(T)$ is either in $\psi(T')$ or has a neighbour $r' \in \psi(T')$, we have $\psi(T) \subseteq N_G[\psi(T')]$.

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 vertex 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 vertices 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 the 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 vertex, so

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

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

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

Proof. We may determine if there exists a mapping ψ with properties stated in Theorem 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 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 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. $\square$

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 1 in time $O(n^{2s+3})$. We prove this claim in Theorem 3. Algorithm 2 is 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. Note that the algorithm in [16] for answering $c_0(G) \leq s$ also runs in time $O(n^{2s+3})$.

Theorem 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

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**

$N_G[v]$ 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 $N_G[T]$ 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. $\square$

3. UPPER BOUNDS FOR $c_k(n)$

Meyniel's conjecture states that $c_0(G) = O(\sqrt{n})$. This conjecture is one of the most difficult unsolved problems regarding the cop number. Finding upper bounds to the cop number is therefore of principal importance, and we address this matter in this section. Our main result in this section is the following upper bound on $c_k(n)$.

Theorem 4. *For integers $n > 0$ and $k \geq 0$ (where k can be a function of n)*

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

From Theorem 4, $c_0(n) = O\left(\frac{n}{\log n}\right)$, which was proven in [9]. We note that recent work of from [22] proved that $c_0(n) = O\left(\frac{n}{\sqrt{\log n}}\right)$.

Before we give the proof of Theorem 4, we consider various lemmas. Fix m a positive integer. We let $N^i_G[T]$ denote $N^i_G[\{T_j : 1 \le j \le m\}]$, for any $T \in V(\boxtimes^t G)$. A homomorphism φ from G to $\boxtimes^m H$, where H is an induced subgraph of G, is called an *m-guarding function* from G to H if

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

Note that $\varphi(x)$ corresponds to an m tuple of vertices of G. Moreover, a subgraph H of G is called *m-guardable* if there is an m-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 m-guarding functions as generalizations of retractions. The proof of the following lemma is immediate.

Lemma 5. *Suppose φ is an m-guarding function from G to H, $x \in V(H)$, and $y \in V(G)$ is a vertex of distance $k \ge 1$ from x. Then there is at least one vertex in $\varphi(y)$ whose distance from x in H is at most k; that is, $x \in N^k_H[\varphi(y)]$.*

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

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

Lemma 6. *For any integer $k \ge 0$ and any m-guardable subgraph H of G,*

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

Proof. Let φ be an m-guarding function from G to H. The strategy for $\mathcal{C}$ is the following: using $c(H) + m - 1$ cops, $\mathcal{C}$ can eventually move, say at round t_0, m 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 m cops will remain at $\varphi(r)$ at all the times $t \ge t_0$, 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 vertex $r \in N_G^{\lfloor \frac{k}{2} \rfloor}[V(H)]$ at round $t > t_0$. Then there is a vertex $x \in V(H)$ of distance $\ell \le \left\lfloor \frac{k}{2} \right\rfloor$ from r. If $\ell \ge 1$, by Lemma 5, 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 vertices 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 m-guarding function, $r = x \in N_H[\varphi(r')]$ and since there are cops in all the vertices 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 vertex 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

$$G[V(G) \setminus N_G^{\lfloor k/2 \rfloor}[V(H)]]$$

using $\max\{\Lambda(k, G, H), c(H) - 1\}$-many cops. □

Lemma 6 says that we can remove the vertices of $\lfloor \frac{k}{2} \rfloor$-neighbourhood of H from G at the cost of at most m cops, that is, $N_G^{\lfloor \frac{k}{2} \rfloor}[H]$ can be "guarded" by m cops. For a given m and k, how large can the closed $\lfloor \frac{k}{2} \rfloor$-neighbourhood of an m-guardable subgraph be? The following lemma answers this question and was implicit in [9].

Lemma 7. *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 vertices of P be $p_1, p_2, \ldots, p_\ell$. For all $1 \leq i \leq 5$ define the homomorphism

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

Then φ s a 5-guarding function from G to H. □

We use Lemma 7 together with the following lemma to obtain 5-guardable subgraphs with large neighbourhoods.

Lemma 8. *For any two integers $n, d \geq 1$ and any rooted n-vertex 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-vertex 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 $n+1$-vertex 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 vertices 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 vertices 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 vertices, $b_{d-1} \geq d$. Let $v \in B_d \setminus B_{d-1}$ be the vertex that maximized the number of vertices 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 $|N^d_{T_v}[P_v]| \geq \tau(\frac{n+1-b_{d-1}}{b_d-b_{d-1}}, d)$. 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

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

□

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.

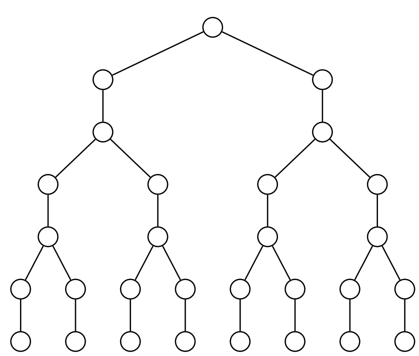

FIGURE 1. A rooted tree showing that $\tau(n,d) \leq 2d\log(1+\frac{n}{d})$, where $n = 29$ and $d = 2$.

With Lemma 8 and Lemma 7 we now may prove Theorem 4.

Proof of Theorem 4. Let G be an n-vertex connected graph and T be a rooted spanning BFS tree of G (see [21] for the definition of BFS trees). By Lemma 8, T has a root-to-leaf path P, such that $|N^d_T[P]| \geq \frac{d\log(1+\frac{n}{d})}{1+\log d}$, 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 7. Since $c(T')=1$, we can use Lemma 6 to obtain that

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

Therefore,

$$\begin{aligned} c_k(n) &= O\left(\frac{n\left(1+\log d\right)}{d\log\left(1+\frac{n}{d}\right)}\right) \\ &= O\left(\frac{n}{\log\left(\frac{2n}{k+1}\right)} \frac{\log(k+2)}{k+1}\right). \end{aligned}$$

□

4. Lower bounds for $c_k(n)$

By considering graphs arising from projective planes it was noted in [26] that

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

With this fact and with our notation, Meyniel's conjecture may be rephrased as

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

We conjecture that for all $k \geq 1$ (where k may tend to infinity with n)

$$c_k(n) = \Theta\left(\left(\frac{n}{k}\right)^{1/2}\right). \tag{4.1}$$

In this section, we establish a lower bound for $c_k(n)$ in terms of n and k, which supports the conjectured lower bound in (4.1). 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 [13].

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 2. For simplicity, we identify vertices of G with corresponding vertices in G^ℓ; in particular, $V(G) \subseteq V(G^\ell)$. Vertices of G^ℓ that are not in G are called *internal vertices*.

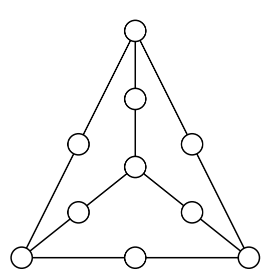

Figure 2. The graph K_4^2.

Lemma 9. *For any graph G and any integer $k \geq 0$,*

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

Lemma 9 sets up a relationship between $c(G)$ and $c_k(G)$. We note that either of the two values bounding $c_k(G^{(2k+1)})$ in the lemma may be realized. For example, $c_1((K_3)^3) = 2$ with $c(K_3) = 1$, while $c(G) = c_k(G^{(2k+1)})$ if G is a tree.

Proof. Joret et al. [20] proved that $c(G^{(2k+1)}) \leq c(G) + 1$. Since $c_k(G^{(2k+1)}) \leq c(G^{(2k+1)})$, it remains to prove that $c(G) \leq c_k(G^{(2k+1)})$.

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

For each internal vertex $x \in V(G^{(2k+1)})$ there is exactly one vertex in $V(G)$ whose distance from x is at most k; name this vertex x_k. Define a function f from the vertices of $G^{(2k+1)}$ to vertices of G that is the identity on $V(G)$, so that if x is internal vertex, then $f(x) = x_k$. The robber $\mathcal{R}$ simulates the winning strategy for Cops and Robbers played on G in distance k Cops and Robbers 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 vertex $r \in V(G)$ 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 vertex 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. 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. $\square$

A result of Fomin et al. [11] states that there is a constant $c > 0$ such that there is no polynomial-time algorithm to approximate $c(G)$ within ratio $c \log n$, unless P=NP. Combining this fact with Lemma 9 gives the following corollary.

Corollary 10. *For any integer $k \geq 0$, computing $c_k(G)$ is NP-hard.*

Proof. Assume that there is an integer k and a polynomial-time algorithm A such that $A(G) = c_k(G)$, for all graphs G. Let B be a polynomial-time algorithm such that $B(G) = G^{(2k+1)}$, for all graphs G. By Lemma 9, it follows that the composition of the algorithms A and B is a polynomial-time 2-approximation algorithm for computing $c(G)$. $\square$

Lemma 9 gives us a tool for transferring lower bounds on $c(n)$ to lower bounds on $c_k(n)$.

Theorem 11. *For all $k \geq 1$ and $n \geq 1$ integers, we have that*

$$c_k(n) \geq \left(\frac{n}{k}\right)^{1/2+o(1)}.$$

Proof. Consider a random graph $G = G(n,p)$ with average degree $np = 3 \log n$. Then a.a.s. G is connected, and by a result from [6] (restated at the beginning of Section 5 below) $c(G) = n^{1/2+o(1)}$ a.a.s. Now by Lemma 9 a.a.s.

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

Since a.a.s. $N = |V(G^{(2k+1)})| = \Theta(k|E(G)|) = kn^{1+o(1)}$, the proof follows since a.a.s.

$$c_k(G^{(2k+1)}) \geq \left(\frac{N}{k}\right)^{1/2+o(1)}.$$

□

5. Random graphs

From [8] it follows that if $p = o(1)$ and $np = n^{\alpha+o(1)}$, where $1/2 < \alpha \leq 1$, then a.a.s.

$$c(G(n,p)) = \Theta(\log n/p) = n^{1-\alpha+o(1)},$$

a.a.s. $c(G(n,p)) = (1+o(1)) \log_{1/(1-p)} n$ for a constant $p < 1$, and a.a.s.

$$c(G(n, n^{-1/2+o(1)})) = n^{1/2+o(1)}.$$

On the other hand, it was proved in [6] that the cop number of $G(n,p)$ is always bounded from above by $n^{1/2+o(1)}$ and this bound is achieved at the other end of the spectrum; that is, for sparse random graphs. More precisely, they showed that $c(G(n,p)) \leq 160000\sqrt{n} \log n$ for $np \geq 2.1 \log n$ and

$$c(G(n,p)) \geq \frac{1}{(np)^2} n^{\frac{1}{2} \frac{\log\log(np)-9}{\log\log(np)}}$$

for $np \to \infty$. Since if either $np = n^{o(1)}$ or $np = n^{1/2+o(1)}$, then a.a.s. $c(G(n,p)) = n^{1/2+o(1)}$, it would be natural to assume that the cop number of $G(n,p)$ is close to $\sqrt{n}$ also for $np = n^{\alpha+o(1)}$, where $0 < \alpha < 1$. In [23] it was shown that the actual behaviour of $c(G(n,p))$ is more complicated. For a fixed integer $k \geq 0$, function $f_k : (0,1) \to \mathbb{R}$ defined as

$$f_k(x) = \frac{\log \mathbb{E}(c_k(G(n, n^{x-1})))}{\log n},$$

where $\mathbb{E}(c_k(G(n,p)))$ denotes the expected value of the distance k cop number for $G(n,p)$. The main result of [23] was that f_0 has an unexpected zigzag shape; see Figure 3.

We actually found zigzags for all $k \geq 0$, as described in the following theorem. See Figure 3 for the functions f_k in the cases $k = 0, 1, 2$.

Theorem 12. *Let $k \geq 0$, $0 < \alpha < 1$, and $d = d(n) = np = n^{\alpha+o(1)}$.*

(1) *If $\frac{1}{2j+1+k} < \alpha < \frac{1}{2j+k}$ for some $j \geq 1$, then a.a.s.*

$$c_k(G(n,p)) = \Theta(d^j).$$

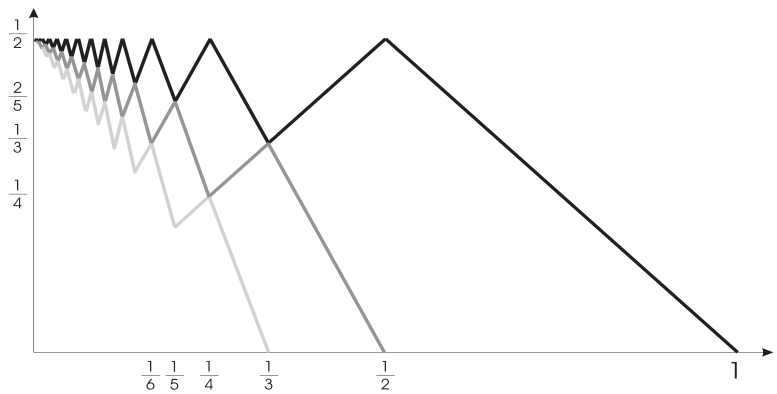

FIGURE 3. The functions f_k, for $k = 0$, 1, and 2, with the darker lines representing smaller values of k.

(2) *If* $\frac{1}{2j+k} < \alpha < \frac{1}{2j-1+k}$ *for some* $j \geq 1$, *then a.a.s.*

$$\Omega\left(\frac{n}{d^{j+k}}\right) = c_k(G(n,p)) = O\left(\frac{n\log n}{d^{j+k}}\right).$$

The proof of Theorem 12 relies on the following two lemmas (essentially from [23]) which supply upper and lower bounds, respectively, for $c_k(G(n,p))$.

Lemma 13. *Let* $k \geq 0$, $j \geq 1$, *and* $d = d(n) = np$.

(1) *If* $n^{1/(2j+1+k)}(\log n)^{1/(j+1+k)} \leq d \leq (n/\log n)^{1/(2j+k)}$, *then a.a.s.*

$$c_k(G(n,p)) = O\big(d^j\big).$$

(2) *If* $(n/\log n)^{1/(2j+2+k)} \leq d \leq n^{1/(2j+1+k)}(\log n)^{1/(j+1+k)}$, *then a.a.s.*

$$c_k(G(n,p)) = O\Big(\frac{n\log n}{d^{j+1+k}}\Big).$$

Lemma 14. *Let* $k \geq 0$, $0 < \alpha < 1$, *and* $d = d(n) = np = n^{\alpha+o(1)}$.

(1) *If* $\frac{1}{2j+1+k} < \alpha < \frac{1}{2j+k}$ *for some* $j \geq 1$, *then there is a constant* $c = c(j,\alpha,k)$ *such that a.a.s.*

$$c_k(G(n,p)) \geq \left[\frac{d}{cj}\right]^j.$$

(2) *If* $\frac{1}{2j+k} < \alpha < \frac{1}{2j-1+k}$ *for some* $j \geq 1$, *then there is a constant* $c = c(j,\alpha,k)$ *such that a.a.s.*

$$c_k(G(n,p)) \geq \left[\frac{d}{cj}\right]^j \frac{n}{cd^{2j+k}}.$$

The proofs of these lemmas follow with minor modifications from the proofs of Lemmas 2.1 and 2.2 in [23]. For this reason, the proofs of the lemmas are omitted. Nevertheless, for completeness, we give a high level overview of the proofs of the lower and upper bounds. In order to derive the upper bound for $c_k(G(n,p))$, the cops use the following strategy. First, distribute the cops uniformly at random. (The number of cops that is required depends on the parameter p.) We show that regardless of the first move of the robber, the cops can move toward the robber so that eventually the robber is surrounded, and is captured after another few moves. The proof relies on Hall's theorem for matchings in bipartite graphs. For the lower bound, we show that regardless of how the cops move, the robber can move keeping all cops within distance at least $k+1$. (Again, the number of cops is a function of p.) Moreover, the robber is able to maintain the property that only a small fraction of all neighbours within distance i (where $i \geq k+1$) are occupied by a cop. This is enough to set up an inductive proof which ensures that the robber can move indefinitely without capture.

Proof of Theorem 12. Theorem 12 follows from Lemmas 13 and 14, along with the consideration of one additional case. The only interval which is not covered in the lemmas is $\frac{1}{2+k} < \alpha < \frac{1}{1+k}$, which we now consider.

In order to get an upper bound, note that the probability that the distance between any pair of vertices v, w is at most $k+1$ is $(1+o(1))\frac{d^{k+1}}{n}$. Therefore, a.a.s. any fixed set of $l = \frac{2n\log n}{d^{k+1}}$ vertices has the property that the distance between this set and any vertex is at most $k+1$. Indeed

$$\begin{aligned} \left(1-\left(1-(1+o(1))\frac{d^{k+1}}{n}\right)^{l}\right)^{n-l} &\geq 1-n\left(1-(1+o(1))\frac{d^{k+1}}{n}\right)^{l} \\ &\geq 1-n\exp(-(1+o(1))2\log n) \\ &= 1-o(1)\,. \end{aligned}$$

Thus, a.a.s. it does not matter where the robber starts the game since he is going to be killed (perhaps, from the distance) in the next round. The upper bound holds.

For the lower bound, we can show that, say, $L = \frac{n}{d^{k+1}}$ cops cannot catch the robber. We show that a.a.s. it does not matter where the cops and the robber are, the robber can always escape to the vertex witch is not reachable from any cop.

Fix S a L-subset of vertices and a vertex u at the distance at least $k+1$ from S. For almost all vertices $x \in V(G) \setminus (S \cup \{u\})$, the probability that a vertex x is adjacent to u and no vertex of S is at the distance at most $k+1$ from x is

$$(1+o(1))\frac{d}{n}\left(1-\frac{d^{k+1}}{n}\right)^{L}.$$

Thus, the probability that no suitable vertex can be found for this particular S and u is

$$\left(1-(1+o(1))\frac{d}{n}\left(1-\frac{d^{k+1}}{n}\right)^{L}\right)^{(1+o(1))n}.$$

Let X be the random variable counting the number of S and u for which no suitable x can be found. We then have that the expected value of X satisfies

$$\begin{aligned}\mathbb{E}(X) &= \binom{n}{k}(n-k)\left(1-(1+o(1))\frac{d}{n}\left(1-\frac{d^{k+1}}{n}\right)^L\right)^{(1+o(1))n} \\ &\leq n^{k+1}\left(1-\Omega\left(\frac{d}{n}\right)\right)^{(1+o(1))n} \\ &= n^{k+1}\exp\left(-\Omega(d)\right) \\ &= o(1).\end{aligned}$$

The proof now follows by Markov's inequality. □

Cop-win graphs, where one cop wins, were structurally characterized in [24, 25]. 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 vertices so that for all $2 \leq j \leq n$, there is a $i < j$ such that

$$N[x_j] \cap \{x_1, x_2, \ldots, x_j\} \subseteq N[x_i] \cap \{x_1, x_2, \ldots, x_j\}.$$

For instance, chordal and bridged graph are cop-win; see [3]. No analogous structural characterization of graphs G satisfying $c_k(G) = 1$, where $k \geq 1$ is a fixed integer, is known.

6. Acknowledgements

We thank Carsten Moldenhauer for helpful discussions.

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, *Random graphs, Second edition*, Cambridge Studies in Advanced Mathematics **73** Cambridge University Press, Cambridge, 200
[6] B. Bollobás, G. Kun, I. Leader, Cops and robbers in a random graph, submitted.
[7] A. Bonato, G. Hahn, C. Wang, The cop density of a graph, *Contributions to Discrete Mathematics* **2** (2007) 133–144.
[8] A. Bonato, P. Prałat, C. Wang, Pursuit-evasion in models of complex networks, *Internet Mathematics* **4** (2009) 419-436.
[9] E. Chiniforooshan, A better bound for the cop number of general graphs, *Journal of Graph Theory* **58** (2008) 45–48.
[10] R. Diestel, *Graph theory*, Springer-Verlag, New York, 2000.
[11] F.V. Fomin, P.A. Golovach, J. Kratochvil, N. Nisse, Pursuing fast robber in graph, *Theoretical Computer Science* **411** (2010) 1167-1181.
[12] F.V. Fomin, D. Thilikos, An annotated bibliography on guaranteed graph searching, *Theoretical Computer Science* **399** (2008) 236–245.
[13] P. Frankl, Cops and robbers in graphs with large girth and Cayley graphs, *Discrete Applied Mathematics* **17** (1987) 301–305.

[14] A.S. Goldstein, E.M. Reingold, The complexity of pursuit on a graph, *Theoretical Computer Science* **143** (1995) 93–112.
[15] G. Hahn, Cops, robbers and graphs, *Tatra Mountain Mathematical Publications* **36** (2007) 163–176.
[16] G. Hahn, G. MacGillivray, A characterization of k-cop-win graphs and digraphs, *Discrete Mathematics* **306** (2006) 2492–2497.
[17] P. Hell, J. Nešetřil, *Graphs and homomorphisms*, Oxford University Press, Oxford, 2004.
[18] W. Imrich, S. Klavzar, *Product graphs-Structure and recognition*, Wiley-Interscience series in discrete mathematics and optimization. Wiley-Interscience, New York, 2000.
[19] S. Janson, T. Łuczak, A. Ruciński, *Random Graphs*, John Wiley and Sons, New York, 2000.
[20] G. Joret, M. Kamiński, D.O. Theis, The cops and robber game on graphs with forbidden (induced) subgraphs, submitted.
[21] D.E. Knuth, *The art of computer programming*, Vol 1, Addison-Wesley, 1997.
[22] L. Lu, X. Peng, On Meyniel's conjecture of the cop number, submitted.
[23] T. Łuczak, P. Prałat, Chasing robbers on random graphs: zigzag theorem, accepted to *Random Structures and Algorithms*.
[24] R. Nowakowski, P. Winkler, Vertex-to-vertex pursuit in a graph, *Discrete Mathematics* **43** (1983) 235–239.
[25] A. Quilliot, Jeux et pointes fixes sur les graphes, Ph.D. Dissertation, Université de Paris VI, 1978.
[26] P. Prałat, When does a random graph have constant cop number?, *Australasian Journal of Combinatorics* **46** (2010) 285-296.
[27] D.B. West, *Introduction to Graph Theory, 2nd edition*, Prentice Hall, 2001.

Department of Mathematics, Ryerson University, Toronto, ON, Canada
E-mail address: abonato@ryerson.ca

Department of Computer Science, University of Western Ontario, London, ON, Canada
E-mail address: ehsan@csd.uwo.ca

Department of Mathematics, West Virginia University, Morgantown, WV 26506-6310, USA
E-mail address: pralat@math.wvu.edu