Zero-Visibility Cops and Robber Game on a Graph

Dariusz Dereniowski[1,⋆], Danny Dyer[2],
Ryan M. Tifenbach[2,⋆⋆], and Boting Yang[3]

[1] Department of Algorithms and System Modeling, Gdańsk University of Technology, Gdańsk, Poland
[2] Department of Mathematics and Statistics, Memorial University of Newfoundland, St. John's, Canada
[3] Department of Computer Science, University of Regina, Regina, Canada

Abstract. We examine the zero-visibility cops and robber graph searching model, which differs from the classical cops & robber game in one way: the robber is invisible. We show that this model is not monotonic. We also provide bounds on both the zero-visibility copnumber and monotonic zero-visibility copnumber in terms of the pathwidth.

1 Introduction

Using mobile agents to find and capture a mobile intruder is a well-studied graph theory problem. Depending on the restrictions placed on the agents and the intruder, the resulting pursuit can vary wildly. One common restriction placed on both the agents and the intruder is a speed limit; in some versions of this game, while the agents may only move along edges one at a time, the intruder may move from any position on the graph to any other along a connected path that does not contain any agents. In other versions, the agents may "jump" from a vertex to any other vertex. In still other games, one or both of the agents and the intruder have limited information about the other party's position; that is, one party or the other may only see the opposition if they are near one another, or alternatively, may never see each other until they stumble upon each other at the same vertex.

The cop and robber model was introduced independently by Winkler and Nowakowski [14] and Quilliot [15]. In this model, a slow, visible intruder (the robber) moves from vertex to adjacent vertex in a graph, while pursued by one slow, visible agent (the cop), who also moves from vertex to adjacent vertex. In these first papers, *copwin* graphs were characterised; that is, graphs where exactly one cop was sufficient to capture. Many questions have grown out of

⋆ D. Dereniowski has been partially supported by Narodowe Centrum Nauki under contract DEC-2011/02/A/ST6/00201.

⋆⋆ Corresponding author. R.M. Tifenbach has been supported by a postdoctoral fellowship from the Atlantic Association for Research in the Mathematical Sciences.

M. Fellows, X. Tan, and B. Zhu (Eds.): FAW-AAIM 2013, LNCS 7924, pp. 175–186, 2013.

these papers. Recently, a characterisation of the *k-copwin* graphs has been discovered [3].

A variation of a less-studied version of this problem dates back to Tošić in 1985 [18]. This corresponds to the cop and robber model with one exception: the robber is invisible. This *zero-visibility* cops and robber model may also be taken as a particular instance of a k-visibility cops and robber problem, where both cops and robber move as in the standard Winkler-Nowakowski-Quilliot model, but the robber only becomes visible to the cops when he is at distance at most k from some cop. In these models, the analog of copnumber can be defined in different ways. Following [17] and [18], we define the zero-visibility copnumber of a graph G to be the minimum number of cops needed to guarantee capture of an invisible robber in a finite time. (Other authors [11] do not necessarily include the restriction to finite time, which works well with their application of the probabilistic method.)

The zero-visibility copnumber for paths, cycles, complete graphs and complete bipartite graphs were characterised in [18], as were graphs that are zero-visibility copwin. There are several constructions in [10] for graphs which require at most 2 cops to perform a zero-visibility search, but a characterisation remains open. An algorithm for determining the zero-visibility copnumber of a tree was given in [17,5], but the problem for general graphs is NP-complete [5]. Most recent work on these topics has been on limited (but not zero) visibility [8,9], and on the expected capture time of the robber in the zero or limited visibility case [1,9,11].

One topic that has been a mainstay of edge searching problems is *monotonicity*. Basically, a search is monotonic if, once a region has been guaranteed to be free of the robber, the cops may not move in such a way to allow the robber to re-enter that region. It is well known that edge searching is monotonic [2,13], but that connected edge searching is not [20,21]. In the original cops and robber game, it was hard to motivate a definition of monotonicity, as the robber was visible. In the zero-visibility version this becomes a natural question again. We will show that the zero-visibility copnumber is different from its monotonic equivalent, and we will discuss bounds on both of these numbers based on the pathwidth of the graph.

2 Zero-Visibility Cops and Robber

We consider a pursuit game on a graph we refer to as *zero-visibility cops & robber*. The game is played on a simple connected graph G between two opponents, referred to as the *cop* and the *robber*. The cop controls the movements of a fixed number of cop pieces and the robber player controls the movement of a single robber piece (we refer to both the players and their pieces as cops and robber). The cop player begins by placing the cops on some collection of vertices of G (more than one cop may occupy a vertex) and his opponent then places the robber on a vertex, unknown to the cop. The players then alternate turns, beginning with the cop; on each player's turn, he may move one or more of his pieces from its current vertex to an adjacent vertex (either player may leave any or even all of his pieces where they are). The game ends with a victory for the cop player if, at any point, the robber piece and a cop piece occupy the same vertex. The robber wins if this situation never occurs. It is important to emphasise that, until he has won, the cop player has no information regarding the robber's position or moves – he cannot see the robber piece until it and a cop occupy the same vertex. On the other hand, the cop may, due to his past moves, gain some knowledge on the possible locations of the robber.

All graphs are assumed to be *simple*; any two vertices are joined by at most one edge and there are no loops (edges from a vertex to itself). We introduce the following terminology regarding this game.

For a graph G, V_G and E_G are the vertex and edge sets of G. We use the symbol $x \sim y$ to represent the fact that x and y are distinct vertices joined by the edge $xy \in E_G$ and the symbol $x \simeq y$ to represent that $x \sim y$ or $x = y$. For each $X \subseteq V_G$, the set $N[X] = \{x \in V : \exists y \in X \text{ such that } x \simeq y\}$ is the *closed neighbourhood* of X. If $X = \{x\}$ is a singleton, we use $N[x]$ rather than $N[\{x\}]$ to represent the closed neighbourhood of x. For $X \subseteq V_G$, the *boundary* of X is the set of vertices adjacent to members of X but not contained in X: $\delta(X) = \{y \notin X : \exists x \in X \text{ such that } x \sim y\} = N[X] \setminus X$.

We will make extensive use of the concept of a *walk* in a graph; however, we give walks additional structure normally not present in their definition. We define a walk in a graph to be a (possibly infinite) sequence of vertices $\alpha = (\alpha(0), \alpha(1), \ldots)$ such that for all $t \geq 0$, $\alpha(t+1) \simeq \alpha(t)$. We use walks to describe cops' and robber's movements; if a walk α corresponds to the positions of a single piece within the game – the vertex $\alpha(0)$ is the starting position of the piece and the vertex $\alpha(t)$ is the location of the piece after its controller has taken t turns.

A *strategy* on G for k cops is a finite set of walks $\mathcal{L} = \{l_i\}_{i=1}^k$, all of the same length T (possibly $T = \infty$). A strategy $\mathcal{L}$ corresponds to a potential sequence of turns by the cop player; each walk $l_i \in \mathcal{L}$ corresponds to the moves of one of the cop pieces. The order of a strategy is the number of cop pieces required to execute it. If a strategy has length $T < \infty$, we might imagine that the cop player forfeits if he hasn't won after T moves. We say that a strategy is *successful* if it guarantees a win by the cop player – a successful strategy results in a win for the cops regardless of the moves made by the robber. Evidently, a strategy $\mathcal{L} = \{l_i\}_{i=1}^k$ of length T is successful if and only if for every walk α of length T in G, there are $l_i \in \mathcal{L}$ and $t < T$ such that $\alpha(t) = l_i(t)$ or $\alpha(t) = l_i(t+1)$. (The robber must be caught at some point, either by moving onto a cop or having a cop move onto it.)

The *zero-visibility copnumber* of a connected graph G is the minimum order $c_0 = c_0(G)$ among successful strategies on G; it is the smallest number of cops required to guarantee capture of the robber.

Typically, in pursuit games of this sort, only finite strategies are considered successful – the robber must be caught in a bounded number of turns. However, the following theorem shows that if we allow infinite strategies in the zero-visibility cops & robber game, any successful strategy (as defined above) will succeed in a bounded amount of time, whether or not the strategy itself is finite.

Theorem 1. *Let G be a graph. Any infinite successful strategy on G may be truncated to obtain a finite successful strategy.*

In light of Theorem 1, we will only consider finite strategies. Moreover, we can recast this game as a *node-search* style problem. Rather than imagine an opponent, we simply keep track of all possible vertices on which the robber piece might be found, via the following construction: (1) Initially, every vertex is marked as *dirty*; (2) a dirty vertex is *cleaned* if a cop piece occupies it; and (3) in between each of the cop's turns, every clean vertex that is unoccupied and adjacent to a dirty vertex becomes dirty.

The dirty vertices are the set of all possible locations of the robber. We refer to the step in between the cop's turns where unoccupied vertices may become dirty as *recontamination*.

Let G be a graph and let $\mathcal{L}$ be a strategy of length T. For each nonnegative integer $t \leq T$, let $\mathcal{L}_t$ be the set of vertices occupied by cops after t turns by the cop player; let $\mathcal{R}_t$ be the set of vertices that are dirty immediately *before* the cop's t-th turn; and let $\mathcal{S}_t$ be the set of vertices that are dirty immediately *after* the cop's t-th turn.

In other words, at the beginning of a t-th turn, $t \geq 1$, the cops occupy the vertices in $\mathcal{L}_{t-1}$ and $\mathcal{R}_t$ are the dirty vertices (possible locations of the robber). Then, the cops move and $\mathcal{L}_t$ becomes the vertex set they occupy, and $\mathcal{S}_t$ becomes the set of vertices that are dirty. After the following robber's move $\mathcal{R}_t$ is the set of dirty vertices.

We define, somewhat arbitrarily, $\mathcal{R}_0 = V$. For $t \geq 0$, the relevant rules of the game imply that $\mathcal{S}_t = \mathcal{R}_t \setminus \mathcal{L}_t$, $\mathcal{R}_{t+1} = N[\mathcal{S}_t] \setminus \mathcal{L}_t$, and $\mathcal{L}_{t+1} \subseteq N[\mathcal{L}_t]$.

A strategy of finite length T is successful if and only if $\mathcal{S}_T$ is empty.

In a pursuit game of this sort, a topic of general interest is that of the monotonicity of strategies. Typically, a strategy is *monotonic* if recontamination never occurs. In this case, such a strategy would have

$$\mathcal{R}_0 \supseteq \mathcal{S}_0 \supseteq \mathcal{R}_1 \supseteq \mathcal{S}_1 \supseteq \ldots \supseteq \mathcal{R}_T \supseteq \mathcal{S}_T,$$

where T is the length of the strategy.

However, consider the following possible activity of a single cop piece. Let xy be an edge and suppose we are attempting to construct a strategy that cleans the graph G. If a single cop moves back and forth between x and y (that is, moves from one to the other every turn), the two vertices x and y are guarded from the robber – if the robber moves onto either while this is occurring he will be caught either immediately or on the next turn.

Considering this activity under the node-search model, although the vertices x and y are possibly being recontaminated over and over, the contamination can never "spread" through them, as they are cleaned before they can possibly recontaminate any further vertices.

We will refer to the above activity as *vibrating* on the edge xy. Further, if E is a set of edges, we say that a set of cops is vibrating on E if each is vibrating on a member of E and every member of E is thus protected. We will also occasionally refer to a set of cops vibrating on a set of vertices X; this simply means that X

is covered by some set of edges E and the cops are vibrating on E. Typically, we want a set of cops to vibrate on a matching (a set of edges that do not share any endpoints), in order to most efficiently utilise this tool.

We wish to take advantage of the above strategic element while still exploring the topic of monotonicity. Thus, we define a strategy of length T to be *weakly monotonic* if for all $t \leq T-1$, we have $\mathcal{S}_{t+1} \subseteq \mathcal{S}_t$. In a weakly monotonic strategy, every time a clean vertex is recontaminated, it is cleaned on the very next move by the cop.

The *monotonic zero-visibility copnumber* of a connected graph G is the minimum order $mc_0 = mc_0(G)$ among successful weakly monotonic strategies on G; it is the smallest number of cops required to capture the robber utilising a weakly monotonic strategy. We are exclusively interested in weakly monotonic strategies as opposed to the stronger variant, and so we will simply use the term monotonic, with the understanding that this means weakly monotonic as defined above. Clearly, we have $c_0(G) \leq mc_0(G)$ for all graphs G.

A *matching* in a graph is a set of edges such that no two are incident (share an endpoint). The *matching number*, $\nu(G)$, is the maximum size of a matching in the graph G. It is well-known that a maximum matching, and thus the matching number, can be found in polynomial time [6].

Theorem 2. *Let G be a connected graph; then, $mc_0(G) \leq \nu(G)+1$, with equality if and only if G is a complete graph on an odd number of vertices.*

A *clique* in a graph is a set of vertices that are all adjacent to each other. The *clique number* of the graph G, denoted as $\omega(G)$, is the maximum of size of a clique in G. A *complete graph* with n vertices, denoted as K_n, is a graph whose clique number is n. Theorem 3 appears in [17,18].

Theorem 3. *If G is a connected graph, then $c_0(G) \geq \frac{1}{2}\omega(G)$. Moreover, $c_0(K_n) = mc_0(K_n) = \lceil \frac{n}{2} \rceil$.*

3 Pathwidth and the Zero-Visibility Copnumber

Let G be a graph with vertex set V_G. A *path decomposition* of G is a finite sequence $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_n)$ of sets $\mathcal{B}_i \subseteq V_G$ such that

1. $\bigcup_{i=1}^{n} \mathcal{B}_i = V_G$;
2. if $x \sim y$, then there is $i \in \{1, \ldots, n\}$ such that $\{x, y\} \subseteq \mathcal{B}_i$; and
3. if $1 \leq i < j < k \leq n$, then $\mathcal{B}_i \cap \mathcal{B}_k \subseteq \mathcal{B}_j$.

We refer to the sets $\mathcal{B}_i$ as *bags*. An alternate, but equivalent, formulation of the third requirement is that for each vertex x, the bags that contain x form a consecutive subsequence, $(\mathcal{B}_i, \mathcal{B}_{i+1}, \ldots, \mathcal{B}_j)$, for some i and j with $1 \leq i \leq j \leq n$.

Let G be a graph and let $\mathcal{B} = (\mathcal{B}_i)$ be a path decomposition of G. We define the *width* of $\mathcal{B}$ to be one less than the maximum size of a bag,

$pw(\mathcal{B}) = \max\{|\mathcal{B}_i| - 1\}$, and the *pathwidth* of G to be the minimum width of a path decomposition of G,

$$pw(G) = \min\{pw(\mathcal{B}) : \mathcal{B} \text{ is a path decomposition of } G\}.$$

The pathwidth of a graph has been introduced in [16].

Lemma 1. *Let G be a connected graph with $pw(G) \leq |V_G| - 2$. Then, there is a path decomposition $\mathcal{B}$ of G containing $n \geq 2$ bags such that $pw(\mathcal{B}) = pw(G)$ and, for each $i = 1, \ldots, n-1$, each of $\mathcal{B}_i \setminus \mathcal{B}_{i+1}$, $\mathcal{B}_{i+1} \setminus \mathcal{B}_i$ and $\mathcal{B}_i \cap \mathcal{B}_{i+1}$ is nonempty.*

The pathwidth of a graph can be characterised via a pursuit game on a graph. Rather than describe the cop and robber dynamics of the game, we will simply examine it as an exercise in cleaning a graph. In this game, the cops do not move along the edges of the graph. Each cop has two moves available to it (although at any point in time only one is possible): (1) if a cop is currently on a vertex in the graph, it may be "lifted" off the graph; and (2) if a cop is currently not in the graph, it may be "placed" on any vertex in the graph. On each of the cop's turns, each of his pieces may make only one move – moving a cop from one vertex to another requires two turns. Initially, every edge is marked as dirty (as opposed to the zero-visibility game, where the vertices are the objects being cleaned). An edge is cleaned when both of its endpoints are occupied by a cop. After each move by the cop player, a clean edge is recontaminated if there is a path joining it to a dirty edge which contains no cops – in this game, the robber moves arbitrarily fast.

This pursuit game is often referred to as *node-searching*; it can be shown that $pw(G) \leq k$ if and only if there is a successful node-search strategy on G utilising $k+1$ cops [7,12].

We introduce this second pursuit game as it is utilised in the proof of Lemma 5; the remainder of this work deals exclusively with the zero-visibility game previously defined.

We produce the following series of inequalities relating the zero-visibility copnumber, the monotonic zero-visibility copnumber and the pathwidth of a graph. In [19], a pursuit game referred to as *strong mixed search* is introduced. It is possible to prove the results in this section utilising the relationships shown therein between strong mixed search and the pathwidth of a graph.

Theorem 4. *Let G be a connected graph containing two or more vertices. Then, $c_0(G) \leq pw(G)$.*

Theorem 5. *Let G be a connected graph; then, $mc_0(G) \leq 2pw(G) + 1$.*

Theorem 6. *Let G be a connected graph; then, $pw(G) \leq 2mc_0(G) - 1$.*

Corollary 1. *Let G be a connected graph on two or more vertices. Then,*

$$c_0(G) \leq pw(G) \leq 2mc_0(G) - 1 \leq 4pw(G) + 1.$$

In Section 4 we provide constructions that in particular prove that the bound in Theorem 4 is tight and the bound in Theorem 5 is tight up to a small additive constant. Moreover, we also use Theorem 3 to argue that the bound in Theorem 6 is tight as well. The formal analysis of these facts is postponed till Section 5.

4 Constructions

We present two constructions of graphs that elicit very interesting relationships between their pathwidth and their zero-visibility copnumbers.

Let G be a graph; the distance between any two vertices x and y is the minimum length of a path joining x and y and is denoted $d_G(x, y)$. So, if $d_G(x, y) = k \geq 1$, then there is a path joining x and y of length k and there are no shorter such paths. If there are no paths joining x and y, the convention is that $d_G(x, y) = \infty$. If H is a subgraph of G, we have $d_H(x, y) \geq d_G(x, y)$ whenever x and y are both present in H. We say that H is an *isometric subgraph* of G if $d_H(x, y) = d_G(x, y)$ whenever x and y are both present in H. Lemma 2 appears in [17].

Lemma 2. *Let G be a graph. If H is an isometric subgraph of G, then $c_0(H) \leq c_0(G)$.*

Let G be a graph; we refer to an edge $e = xy$ as a *cut edge* if the graph $G \setminus e$ obtained by deleting e (without deleting either of the endpoints x or y) is disconnected. Clearly, if e is a cut edge, then $G \setminus e$ has two connected components. Not every graph contains cut edges.

Lemma 3. *Let G be a graph that contains a cut edge e. If H is one of the connected components of $G \setminus e$, then*

$$c_0(H) \leq c_0(G) \text{ and } mc_0(H) \leq mc_0(G).$$

Moreover, let $\mathcal{L}$ be a successful strategy on G. Then, at some point in the strategy at least $c_0(H)$ cops are simultaneously present in H; if $\mathcal{L}$ is a monotonic strategy, at some point at least $mc_0(H)$ cops are simultaneously present in H.

Corollary 2. *Let G be a tree. If H is a subtree of G, then $mc_0(H) \leq mc_0(G)$.*

Example 1. We produce an interesting example of a graph G with an isometric subgraph H such that $mc_0(G) < mc_0(H)$. This illustrates that Lemmas 2 and 3 and Corollary 2 are limited in how they might be extended.

The graph G in question (see Figure 1) contains a large number of vertices with degree 2; to simplify the depiction, some of them are omitted. Specifically, the two dashed lines are paths of length 8 – they each contain 7 internal degree-2 vertices which are not shown. The subgraph H is obtained by deleting the two paths of length 8 (drawn with dashed lines) and all 7 of their internal vertices.

First, we claim that 2 cops can clean the entire graph in a monotonic fashion. One of them is initially placed on y and stays idle during the entire strategy.

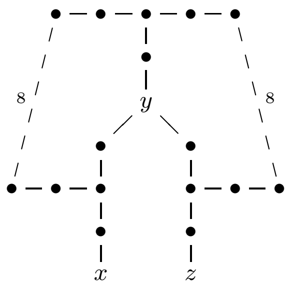


Fig. 1. The subgraph obtained by deleting the two paths of length 8 has strictly higher monotonic zero-visibility copnumber than the supergraph

The other cop is initially placed on x and moves around the remainder of the graph ending on z and cleaning every other vertex. This cop moving around the perimeter cleans the other three vertices adjacent to y – when it reaches a degree-3 vertex it moves onto the neighbour of y and then back.

However, if we delete the two paths drawn as dashed lines, we cannot clean the subgraph monotonically with only two cops – this can be shown in a manner very similar to the proof of Lemma 4.

A *rooted tree* is a tree G where a single vertex has been marked as the *root*. In a rooted tree with root r every vertex $x \neq r$ has a unique parent. The parent of x is identified in the following manner: every vertex $x \neq r$ is joined to r by a unique path – the parent of x is the sole neighbour of x in this path. If y is the parent of x, then x is a child of y; we also use the similarly defined terms grandparent and grandchild when discussing rooted trees.

Example 2. The following family of trees illustrates the distinction between the pathwidth of a tree and its monotonic zero-visibility copnumber. Let T_k be obtained by beginning with the full rooted binary tree of height k and subdividing every edge exactly once. We draw the root of each T_k with a circle – see Figure 2 for the first three such trees. By Lemmas 4 and 5, we have

$$mc_0(T_k) = k \text{ and } pw(T_k) = c_0(T_k) = \left\lfloor \frac{k}{2} \right\rfloor + 1.$$

The recursive Algorithm 1 cleans T_k with a monotonic strategy (this can be shown simply via induction) that utilises $k+1$ cops and begins with every cop placed on the root. However, this is not an optimal strategy – if $k \geq 1$, there is a successful monotonic strategy on T_k using k cops. We obtain this strategy by using Algorithm 1 as follows. We first express T_k as two copies of T_{k-1} joined by a path of length 4:

Algorithm 1. CLEAN(T_k)

Require: There are $k+1$ cops on the root of the graph T_k.

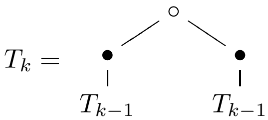
if $k = 0$ **then**

  return

end if

1. Let x be the root of T_k and let y and z be the two grandchildren of x. Let T^y and T^z be the two copies of T_{k-1} rooted at y and z, respectively.
2. In two moves, move $k-1$ of the cops from the root of T_k to y, leaving the remaining cop on x.
3. CLEAN(T^y).
4. Move all $k-1$ cops from T^y to z (each cop does not move any further into T^z).
5. CLEAN(T^z).

return



We clean one copy of T_{k-1} by using Algorithm 1 in such a way that one cop remains on the root of this subtree during the entire strategy. All k cops then move to the root of the other copy of T_{k-1} and clean that subtree in a similar fashion. Thus, $mc_0(T_k) \leq k$, if $k \geq 1$. In Lemma 4, we show that this strategy is, in fact, optimal.

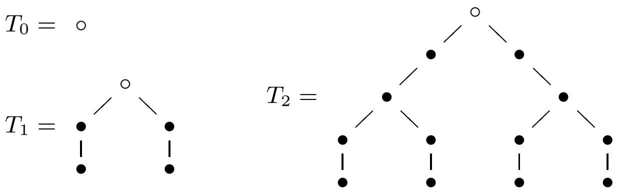


Fig. 2. The first three subdivided binary trees

Lemma 4. *For* $k \geq 1$, $mc_0(T_k) = k$.

Theorem 4 and Lemma 4 together imply that $pw(T_k) \leq 2k-1$. However, we in fact have $pw(T_k) = \left\lfloor \frac{k}{2} \right\rfloor + 1$.

Lemma 5. *For* $k \geq 1$, $pw(T_k) = c_0(T_k) = \left\lfloor \frac{k}{2} \right\rfloor + 1$.

We produce a construction of graphs with $c_0 < pw$ and $c_0 < mc_0$ with the unbounded ratios in both inequalities.

A *universal vertex* in a graph G is a vertex adjacent to every other vertex. Given a graph G, we form the graph G^* by adding a universal vertex to G – that is, a single new vertex is added together with edges joining this new vertex and every other vertex already present in G.

A *subdivision* of a graph G is a graph H formed by replacing one or more edges in G with paths of length greater than or equal to 2; a subdivision is formed by dividing an edge into two or more new edges. If the vertices of G are labeled and H is a subdivision of G, we preserve the labeling of the vertices, adding new labels to the new vertices.

Lemma 6. *If G is a tree containing two or more vertices, then there is a subdivision H of G such that $c_0(H^*) = 2$.*

5 Comparisons between the Zero-Visibility Copnumbers and the Pathwidth of a Graph

We examine in detail the inequality presented in Corollary 1:

$$c_0(G) \leq pw(G) \leq 2mc_0(G) - 1 \leq 4pw(G) + 1.$$

A *caterpillar* is a tree such that deleting all vertices of degree 1 results in a path or an empty graph. The proof of Theorem 7 is a straightforward exercise and is omitted.

Theorem 7. *Let G be a graph. The following are equivalent:*

1. *We have $c_0(G) = 1$, $mc_0(G) = 1$ or $pw(G) = 1$.*
2. *We have $c_0(G) = mc_0(G) = pw(G) = 1$.*
3. *We have $c_0(G) = pw(G) = 2mc_0(G) - 1$.*
4. *The graph G is a caterpillar.*

The class of graphs that minimizes the zero-visibility copnumbers is identical to the class that minimises pathwidth. However, if $c_0(G) = 2$, this gives us absolutely no information concerning $mc_0(G)$ or $pw(G)$, as we will see in Theorem 8.

Remark 1. The bound $c_0(G) \leq pw(G)$ in Theorem 4 is the best possible.

The class of graphs $\{T_k\}$, described in the previous section, satisfy $c_0(T_k) = pw(T_k) < mc_0(T_k)$ (by Lemmas 4 and 5). Thus, the bound $c_0(G) \leq pw(G)$ is tight on an infinite family of graphs and we cannot sandwich mc_0 between c_0 and pw, in general.

Remark 2. The bound $mc_0(G) \leq 2pw(G)+1$ in Theorem 5 can only be improved by a small constant, if it can be improved at all.

The subdivided binary trees T_k described in Example 2 have $mc_0(T_k) = k$ and $pw(T_k) = \lfloor \frac{k}{2} \rfloor + 1$. So,

$$mc_0(T_k) = \begin{cases} 2pw(T_k) - 1 \text{ if } k \text{ is odd, or} \\ 2pw(T_k) - 2 \text{ if } k \text{ is even.} \end{cases}$$

The coefficient of 2 cannot be reduced; only the additive constant can be changed, possibly by reducing it by one or two. In a roundabout manner, this shows that the result in [4] is close to optimal.

Remark 3. The bound $pw(G) \leq 2mc_0(G) - 1$ in Theorem 6 is the best possible.

The complete graph on an even number of vertices has

$$pw(K_{2m}) = 2mc_0(K_{2m}) - 1 = 2m - 1.$$

As well, Theorem 8 shows that we cannot replace Theorem 6 with the stronger statement "$pw(G) \leq 2c_0(G) - 1$". In fact, pathwidth cannot be bounded above by any function of the zero-visibility copnumber.

Theorem 8. *For any positive integer k, there is a graph G with $c_0(G) = 2$ and $pw(G) \geq k$.*

6 Conclusion

There remains a considerable amount of further work concerning the zero-visibility model to be accomplished. Characterisations of c_0 and mc_0 over well-known families of graphs (such as trees, unicyclic graphs, planar graphs, series parallel graphs, *etc.*) are of interest. An analysis of the algorithmic complexity of accomplishing a successful zero-visibility search would cement this model's position in the overall area of pursuit games and width parameters. It would be very interesting to construct some sort of relationship between the value $mc_0(G) - c_0(G)$ (or possibly $mc_0(G)/c_0(G)$) and combinatoric or connective properties of the graph – that is, to answer the question, given some known property of the graph, can we bound the amount by which c_0 and mc_0 differ?

The fact that the monotonic zero-visibility copnumber can be bounded both above and below by positive multiples of the pathwidth suggests that, in a sense, node-search and the monotonic zero-visibility search are variations of the same game – each number is an approximation of the other, suggesting that efficient strategies in one game can usually be translated to efficient strategies in the other.

However, Theorem 8 shows that the zero-visibility copnumber can be entirely unrelated to the pathwidth and the monotonic zero-visibility copnumber. The general zero-visibility search can be carried out using methods that will not work in a node-search – the zero-visibility search is genuinely distinct from other pursuit games and informs us of different structural properties of a graph.

References

1. Adler, M., Räcke, H., Sivadasan, N., Sohler, C., Vöcking, B.: Randomized pursuit-evasion in graphs. Combinatorics, Probability and Computing 12, 225–244 (2003)
2. Bienstock, D., Seymour, P.: Monotonicity in graph searching. Journal of Algorithms 12, 239–245 (1991)
3. Clarke, N.E., MacGillivray, G.: Characterizations of k-copwin graphs. Discrete Mathematics 312, 1421–1425 (2012)
4. Dereniowski, D.: From pathwidth to connected pathwidth. SIAM Journal on Discrete Mathematics 26, 1709–1732 (2012)
5. Dereniowski, D., Dyer, D., Tifenbach, R., Yang, B.: The zero-visibility copnumber of a tree (2013)
6. Edmonds, J.: Paths, trees, and flowers. Canadian Journal of Mathematics 17, 449–467 (1965)
7. Ellis, J.A., Sudborough, I.H., Turner, J.S.: The vertex separation and search number of a graph. Information and Computing 113, 50–79 (1994)
8. Isler, V., Kannan, S., Khanna, S.: Randomized pursuit-evasion with local visibility. SIAM Journal on Discrete Mathematics 20, 26–41 (2006)
9. Isler, V., Karnad, N.: The role of information in the cop-robber game. Theoretical Computer Science 399, 179–190 (2008)
10. Jeliazkova, D.: Aspects of the cops and robber game played with incomplete information. Master's thesis, Acadia University (2006)
11. Kehagias, A., Mitche, D., Prałat, P.: The role of visibility in the cops-robber game and robotic pursuit/evasion (2012)
12. Kinnersley, N.G.: The vertex separation number of a graph equals its path-width. Information Processing Letters 42, 345–350 (1992)
13. LaPaugh, A.S.: Recontamination does not help to search a graph. Journal of the ACM 40, 224–245 (1993)
14. Nowakowski, R., Winkler, P.: Vertex-to-vertex pursuit in a graph. Discrete Mathematics 43, 235–239 (1983)
15. Quilliot, A.: Problèmes de jeux, de point fixe, de connectivité et de représentation sur des graphes, des ensembles ordonnés et des hypergraphes. PhD thesis, Université de Paris VI (1983)
16. Robertson, N., Seymour, P.D.: Graph minors. I. Excluding a forest. Journal of Combinatorial Theory, Series B 35, 39–61 (1983)
17. Tang, A.: Cops and robber with bounded visibility. Master's thesis, Dalhousie University (2004)
18. Tošić, R.: Inductive classes of graphs. In: Proceedings of the Sixth Yugoslav Seminar on Graph Theory, pp. 233–237. University of Novi Sad (1985)
19. Yang, B.: Strong-mixed searching and pathwidth. Journal of Combinatorial Optimization 13, 47–59 (2007)
20. Yang, B., Dyer, D., Alspach, B.: Sweeping graphs with large clique number. In: Fleischer, R., Trippen, G. (eds.) ISAAC 2004. LNCS, vol. 3341, pp. 908–920. Springer, Heidelberg (2004)
21. Yang, B., Dyer, D., Alspach, B.: Sweeping graphs with large clique number. Discrete Mathematics 309, 5770–5780 (2009)