SIAM J. DISCRETE MATH.
Vol. 23, No. 1, pp. 349–368

GRAPH SEARCHING IN A CRIME WAVE*

DAVID RICHERBY† AND DIMITRIOS M. THILIKOS‡

Abstract. We define helicopter cops and robber games with multiple robbers, extending previous research, which considered only the pursuit of a single robber. Our model is defined for robbers that are visible (their position in the graph is known to the cops) and active (they can move at any point in the game) but is easily adapted to other variants of the single-robber game that have been considered in the literature. We show that the game with many robbers is nonmonotone: that is, fewer cops are needed if the robbers are allowed to reoccupy positions that were previously unavailable to them. As the moves of the cops depend on the position of the visible robbers, strategies for such games should be interactive, but the game becomes, in a sense, less interactive as the initial number of robbers increases. We prove that the main parameter emerging from the game, which we denote $\mathbf{mvams}(G, r)$, captures a hierarchy of parameters between proper pathwidth and proper treewidth, and we completely characterize it for trees, extending analogous existing characterizations of the pathwidth of trees. Moreover, we prove an upper bound for $\mathbf{mvams}(G, r)$ on general graphs and show that this bound is reached by an infinite class of graphs. On the other hand, if we consider the robbers to be invisible and lazy, the resulting parameters collapse in all cases to either proper pathwidth or proper treewidth, giving a further case where the classical equivalence between visible, active robbers and invisible, lazy robbers does not hold.

Key words. graph searching, treewidth, pathwidth

AMS subject classifications. 05C83, 05C85

DOI. 10.1137/070705398

1. Introduction.

During recent decades, the problem of searching a graph has attracted much attention not only because of its purely graph-theoretic interest but also for its numerous applications in modeling problems in communication networks (for related surveys, see [1, 2]). In general, graph searching problems are described in terms of a game played between a team of cops and a robber, whom the cops attempt to capture by moving systematically through the graph. We wish to know the minimum number of cops required to catch the robber, subject to various constraints on their behavior and that of the robber. Several versions of the game have been examined, differing, for example, in whether the cops know the position of the robber, whether the robber can move at will or only when disturbed by a cop, and how the cops can move through the graph.

One of the main models of graph searching, known as the helicopter cops and robber game, was introduced by Seymour and Thomas [11]. In this model, the robber occupies a vertex of a graph and is *active* in the sense that he may move at every round of the game along any path of any length whose vertices are not guarded by the cops. On the other hand, the cops are not constrained to stay within the graph and can be placed on or removed from vertices of the graph, as if flying by helicopter.

*Received by the editors October 15, 2007; accepted for publication (in revised form) September 29, 2008; published electronically January 7, 2009. This research carried out while the first author was a post-doc of the Graduate Program in Logic, Algorithms, and Computation ($\mu\prod\lambda\forall$) at the Department of Mathematics, National and Kapodistrian University of Athens.

http://www.siam.org/journals/sidma/23-1/70539.html

†School of Computing, University of Leeds, Leeds, LS2 9JT, UK (richerby@comp.leeds.ac.uk).
This author was funded by the European Social Fund and Greek National Resources (ΕΠΕΑΕΚ II) PYTHAGORAS II.

‡Department of Mathematics, National and Kapodistrian University of Athens, Panepistimioupolis, GR-15784 Athens, Greece (sedthilk@math.uoa.gr).

A crucial feature of this game is that the robber is *visible*: the cops have complete knowledge of his current position. Victory for the cops is declared when a cop lands on the vertex occupied by the robber and the robber cannot make any move to escape. Since the cops base their moves on the current position of the robber, the strategy they use is *interactive*. In [11], Seymour and Thomas proved that the minimum number of cops guaranteed to be able to win the game is one greater than the treewidth of the graph on which it is played. The proof of this result includes a proof of the monotonicity of the game, i.e., that the cops do not become weaker when their moves are restricted to those that monotonically decrease the portion of the graph available to the robber.

Variants of the above game were considered in [6], where now the robber is a *lazy* fugitive who moves only when a cop lands on the vertex he occupies. However, to compensate, the robber is now *invisible*: his position is unknown to the cops. Notice that in this game, the cops' strategy is predetermined and can be given in advance. Games defined with this characteristic are described as *fugitive games* in order to stress the invisibility of the robber. This second version is equivalent to the Seymour–Thomas game in the sense that, for any graph, the two games require the same number of cops [6]. It follows easily from [6] (see also [3]) that, when the fugitive is active and invisible, the number of cops required to ensure his capture is one greater than the pathwidth of the graph—another graph parameter of equal importance to treewidth.

This paper intends to examine, and also unify, the above models under the natural extension where the graph contains many robbers rather than just one. This is the first time that multiple robbers have been considered in graph searching, and we believe that our results will motivate such a study for other models as well.

We describe our model for graph searching using the most general setting of *mixed searching*, proposed by Bienstock and Seymour [3] and also examined in [12, 13, 14, 15]. In this model, each move of the cops consists either of a placement or removal (as before) or of *sliding* a cop along an edge of the graph. This may reduce by one the number of cops required to search a graph, but, as observed in [3], the version without sliding can be reduced to a mixed search by replacing each edge in the graph with two parallel edges (or a triangle involving a new vertex). Moreover, apart from being more general, including sliding in our model makes the presentation of our results cleaner.

It is not obvious how to generalize the concept of monotonicity to the setting with many robbers. Now, each robber has his own individual free space, leading to the question of whether monotonicity should be defined individually or collectively. We give three natural definitions and show them to be equivalent.

Monotonicity is crucial in the multiple-robber case. If we do not require monotonicity, we can catch any number r of visible, active robbers one at a time by repeating the strategy to catch a single robber, without requiring any additional cops. However, when we restrict our attention to monotone strategies, the number of cops required, which we denote $\mathbf{mvams}(G, r)$ (for monotone, visible, active, mixed search number against r robbers), can be greater than the nonmonotone case and depends on the number of robbers. In particular, $\mathbf{mvams}(G, 1)$ is just the mixed search number for a single visible active robber. This, in turn, is equal to the parameter of *proper treewidth* defined in [5,12]. On the other hand, if n is the number of vertices in G, then $\mathbf{mvams}(G, n)$ is equal to the mixed search number for a single invisible active robber which, in turn, corresponds to the parameter of *proper pathwidth* defined in [14]. Moreover, we show that $\mathbf{mvams}(G, r)$ can, for appropriate values of r, take all intermediate values between proper treewidth and proper pathwidth. As our main result, we give the exact value of $\mathbf{mvams}(G, r)$ on trees and an upper bound for general graphs:

$$\begin{aligned} &\mathbf{mvams}(T,r) = \min\{\mathbf{ppw}(T), \lfloor \log r \rfloor + 1\} && \text{(for any tree } T\text{)},\\ &\mathbf{mvams}(G,r) \le \min\{\mathbf{ppw}(G), \mathbf{ptw}(G)\cdot(\lfloor \log r \rfloor + 1)\} && \text{(for any graph } G\text{)}, \end{aligned}$$

where $\mathbf{ppw}(G)$ and $\mathbf{ptw}(G)$ denote, respectively, the proper pathwidth and proper treewidth of the graph G.

Our result for trees is based on a complete characterization of $\mathbf{mvams}(T,r)$ on trees and extends the analogous characterizations for pathwidth and proper pathwidth given in [13] and [7], respectively.

Our results can be seen as showing that the number of robbers tunes the amount of interactivity in search strategies, spanning all intermediate levels from pathwidth (fully predetermined) to treewidth (fully interactive). A rather different way of defining this tuning was given by Fomin, Fraigniaud, and Nisse, who considered a single active robber but restricted the number of rounds at which the cops can ask for the robber's position [8].

A natural question is whether the same variation of values can be achieved in the setting of invisible but lazy fugitives defined in [6], given that a single invisible lazy fugitive is equivalent to a single visible active robber. However, in the case of multiple robbers, being lazy and invisible is not the same as being active and invisible. Here, we can define laziness as meaning either that a robber may move only when the cops land on his vertex or that all robbers may move together when a cop lands on any single robber. For either definition of laziness, with or without monotonicity, the hierarchy collapses, and, in a graph of order n, any number of robbers is equivalent to either a single robber or n robbers. Thus, the multiple-robber setting is degenerate for games with predetermined strategies, which supports our decision to consider the interactive strategies generated by the visible, active setting. Note that this is not the first case where invisibility cannot be exchanged for laziness: Hunter and Kreutzer have shown that the symmetry breaks, even for one robber, when the games are defined on directed graphs [10].

The remainder of the present paper is organized as follows. Our graph searching model is defined in detail in section 2. In section 3, we show the equivalence of three reasonable definitions of monotonicity and explore the role of monotonicity in the game. To relate our hierarchy of parameters to the well-known parameters of proper pathwidth and proper treewidth, we make a brief detour through the theory of games with an invisible robber in section 4, where we also show that the case of multiple invisible robbers collapses to already-studied cases. In section 5, we give upper bounds for the number of cops required to catch r robbers in trees and in general graphs, and, in section 6, we show that the upper bound for trees is, in fact, an exact characterization of the number of cops needed. We also show that the upper bound for general graphs is reached by an infinite class of graphs. Several consequences and open problems emerging from our results are presented in section 7.

2. The searching model. All graphs considered in this paper are finite, simple, and, unless otherwise stated, undirected.

In a helicopter search game with many visible robbers, the opponents are a group of k cops and a group of r robbers, who occupy vertices of the graph. The goal of the cops is to capture all of the robbers. At all times, the cops and robbers have full information about each other's location and may use this information to decide their next move. Initially, there are no cops in the graph, but, at all times, any robber who has not been captured is on some vertex.

A play of the game consists of a sequence of rounds, with each round consisting of three parts, as follows.

Announcement. The cops announce their intended move to the robbers. One cop moves in each round, by one of the following operations.

- Placement of a cop on a vertex v, not currently occupied by a cop. The move is denoted by **place**(v).
- Removal of a cop from an occupied vertex v, denoted by **remove**(v).
- Sliding of a cop from the one endpoint u of an edge $\{u, v\}$ to the other, which is initially not occupied by a cop. The move is denoted by **slide**$(u \to v)$.

Avoidance. Each robber who has not yet been captured can move with infinite speed to any vertex reachable from his current position by a path not blocked by cops, as long as this vertex will not be occupied by a cop once the cops' current move has been realized. The robbers are "active" in the sense that any robber may move in the graph at any move of the game, as long as he has an unblocked path to move along.

If the announced move is a placement to or removal from some vertex, that vertex is not considered to be blocked for the purposes of the robbers' movement in the round. If the announced move is **slide**$(u \to v)$, the edge uv is considered to be blocked for this round but the vertices u and v are not.

Realization. The cops carry out the announced action.

A robber is captured if the cops announce that they will move (by placement or sliding) to the vertex he occupies and there is no way for him to move to another vertex.

To formalize the game, we will use a string $\mathbf{R} \in (V(G) \cup \{*\})^r$ to denote the positions of the r robbers in the graph. In particular, the ith character of $\mathbf{R}$ is either the vertex occupied by the ith robber or "$*$" in the case that the ith robber has been captured. We write $V(\mathbf{R})$ for the set of characters in $\mathbf{R}$, other than $*$. Since, at any time, there is at most one cop on any vertex, we may represent the position of the cops as a set $S \in V(G)^{[\leq k]}$.

A *play* of the game on a graph is an infinite sequence of positions

$$\mathcal{P} = S_0, \mathbf{R}_0, S_1, \mathbf{R}_1, \dots ,$$

where, for each i, the transition from having the cops at S_i and robbers at $\mathbf{R}_i$ to the cops at S_{i+1} and robbers at $\mathbf{R}_{i+1}$ is a valid move of the game, as described above. Specifically, the sequence $S_0, S_1, \dots$ has the properties that

- $S_0 = \emptyset$;
- $S_1 = \{v\}$ for some vertex v—the first move is **place**(v); and
- for consecutive sets S_i and S_{i+1}, one of the following holds:
  - $S_{i+1} - S_i = \{v\}$—the move is **place**(v),
  - $S_i - S_{i+1} = \{v\}$—the move is **remove**(v),
  - $S_{i+1} \triangle S_i = \{u, v\} \in E(G)$—the move is **slide**$(u \to v)$, where $S_i - S_{i+1} = \{u\}$ and $S_{i+1} - S_i = \{v\}$.

We call such a sequence of cop positions *consistent.*

Given two consecutive sets S_i and S_{i+1} of a consistent sequence, we say that a path P of G is (S_i, S_{i+1})-*avoiding* if its internal vertices avoid $S_i \cap S_{i+1}$, its last vertex is not in S_{i+1}, and, in the case that $|e| = 2$, its edges avoid the edge $e = S_{i+1} \triangle S_i$.

Given that the location of the robbers at the ith step is $\mathbf{R}_i = [a_1 \dots a_r]$, we define the set of free locations for the jth robber after step i as $F^j_{i+1} = \emptyset$ if $a_j = *$ and, otherwise,

$$F^j_{i+1} = \{y \in V(G) - S_{i+1} \mid G \text{ contains an } (S_i, S_{i+1})\text{-avoiding } (a_j, y)\text{-path}\}.$$

As a response to the ith move of the cops, the robbers can choose their new location to be any string $\mathbf{R}_{i+1} = [a'_1 \dots a'_r]$ such that for $j \in \{1, \dots, r\}$, $a'_j = *$ if $F^j_{i+1} = \emptyset$ and $a'_j \in F^j_{i+1}$ otherwise. (In particular, note that, if $a_j = *$, then $a'_j = *$ also.)

We set $F_0 = V(G)$, and for $i \geq 1$, we define $F_i = \bigcup_{j \in \{1,\dots,r\}} F^j_i$. We say that the sequence $F_0, F_1, \dots$ is the sequence of *free positions* for the robbers. If, for every $i \geq 0$, $F_{i+1} \subseteq F_i$, we say that $\mathcal{P}$ is a *monotone* play. (Other definitions of monotonicity are considered in section 3 and shown to be equivalent to this definition in the sense that they lead to the same graph parameter.)

A play $\mathcal{P} = S_0, \mathbf{R}_0, S_1, \mathbf{R}_1, \dots$ is *winning* (for the cops) if $V(\mathbf{R}_i) = \emptyset$ for some $i \geq 0$; that is, all the robbers are eventually captured. The *essential part* of a winning play is the subsequence $S_0, \mathbf{R}_0, \dots, S_\ell, \mathbf{R}_\ell$, where ℓ is minimal such that $V(\mathbf{R}_\ell) = \emptyset$.

According to our description of the game, any move of the cops may depend on the current position of the cops and robbers in the graph. A *search strategy of cost k against r robbers* or, more succinctly, a *(k, r)-strategy* is a function

$$\mu : V(G)^{[\leq k]} \times (V(G) \cup \{*\})^r \to V(G)^{[\leq k]},$$

whose inputs are the position S of the cops and the positions $\mathbf{R}$ of the robbers and whose output is S', the new position of the cops, such that, for all S and $\mathbf{R}$, the sets S and S' obey the restrictions given in the definition of consistency for sequences. That is, there is a single move which transforms the cop position S to S'.

Note that, when we define strategies, we will not define the action of the cops in positions that can never occur when the strategy is executed. Thus, we give only a partial function. Formally, the strategy is any total extension of this partial function, assigning arbitrary moves to the cops in situations that do not occur in any play in which the cops follow the given partial strategy.

A *play with respect to a (k, r)-strategy* μ, or a *μ-play*, is any play $S_0, \mathbf{R}_0, S_1, \mathbf{R}_1, \dots$ where $S_{i+1} = \mu(S_i, \mathbf{R}_i)$ for all $i \geq 0$. A strategy μ is said to be *monotone* if all μ-plays are monotone and *winning* if all μ-plays are winning.

We define the *nonmonotone* and *monotone visible active mixed search number*, respectively, of a graph G as follows:

$$\mathbf{vams}(G, r) = \min \{k \mid \text{there is a winning } (k, r)\text{-strategy on } G\},$$
$$\mathbf{mvams}(G, r) = \min \{k \mid \text{there is a monotone winning } (k, r)\text{-strategy on } G\}.$$

To describe a search strategy as even a partial function is often rather cumbersome. Instead, we will frequently describe a search strategy as a *search program* Π that makes move decisions depending only on the current position of the cops and robbers, without reference to previous positions in the search. Thus, we can extract a strategy from a search program and vice versa. We call a search program monotone or winning if the corresponding search function is. The program receives the information on the positions of the robbers by calling a routine *robbers_positions()*.

As an example we give program 1, a monotone winning search program for one cop against one robber in a tree T. Notice that, at each step, the robber must choose

Search program 1. $\Pi(T, 1)$ to capture one robber in a tree T.

place(v) where v is any vertex of T.
Let $\mathbf{R} \leftarrow$ *robbers_positions()*.
Let $T' \leftarrow T$.
While $V(\mathbf{R}) \neq \emptyset$,
    Let T' be the connected component of $T - v$ containing $V(\mathbf{R})$
        and let w be the (unique) vertex of T' adjacent to v.
    slide($v \rightarrow w$).
    Let $v \leftarrow w$.
    Let $\mathbf{R} \leftarrow$ *robbers_positions()*.
remove(v).

his position in the connected component T' where he resides, excluding the vertex w that is the target of the cop's move. At each round, the set of free positions of the robber becomes strictly smaller, ensuring both monotonicity and the eventual capture of the robber.

We may also represent a winning (k, r)-strategy μ as a finite tree. Let T_μ be the least labeled, rooted, directed tree with the following properties. (By "least," we mean that no proper subtree of T_μ can be labeled to meet our requirements. Note that there may be more than one vertex or edge with any given label and that, when we speak of a path in T_μ, we mean a maximal directed path from the root to a leaf. We could also represent nonwinning strategies as infinite trees in a similar way, but we need only representations of winning strategies.)

- Every edge is directed away from the root.
- Every vertex is labeled with a set $S \in V(G)^{[\leq k]}$, and every edge is labeled with a string $\mathbf{R} \in (V(G) \cup \{*\})^r$.
- The essential part of every μ-play in G labels some path in T_μ.

Notice that, according to the above, the root of T_μ is labeled by the empty set, corresponding to the position of the cops at the beginning of any μ-play.

Our manipulation of tree representations of strategies will often lead us to construct trees that do not represent strategies because they are *defective* in some way. Allowing such trees makes several of our proofs more straightforward. Here, we describe the defects that may arise and show how to repair them.

Nondeterminism. A vertex v labeled S might have distinct outgoing edges vv' and vv'' with the same label $\mathbf{R}$ but with v' and v'' having labels S' and S'' (where S' and S'' are not necessarily distinct). Thus, with the cops in position S and the robbers at $\mathbf{R}$, the cops can win by moving to either S' or S''. Hence, we may delete the subtree rooted at v''.

Null moves. A vertex v labeled S might have a child v' also labeled S. This corresponds to the cops deciding to do nothing for a move. Since the robbers may move anywhere in their free space, allowing them to make two consecutive moves while the cops stay still gives them no extra power. Hence, we may delete the vertex v' and, for every child w of v', where the edge $v'w$ is labeled $\mathbf{R}$, add an edge vw, also labeled $\mathbf{R}$.

Inconsistency. There may be distinct vertices v and w, both labeled S, with outgoing edges vv' and ww', respectively, that have the same label $\mathbf{R}$ but with v' and w' having distinct labels S' and S'', respectively. As in the case of nondeterminism, this means that the cops have a choice of ways to win from the position $(S, \mathbf{R})$. We may replace the subtree rooted at w' with a copy of the subtree rooted at v'.

Repeated application of these operations will yield a tree that properly corresponds to a winning strategy. Further, the resulting strategy uses no more cops than were deployed in the original defective tree and is monotone if and only if every play in the defective tree was monotone.

Note, in particular, that the discussion above of inconsistent trees justifies our decision to define strategies as functions:

$$\mu : V(G)^{[\leq k]} \times (V(G) \cup \{*\})^r \to V(G)^{[\leq k]}.$$

Such strategies are known as *positional* or *memoryless* strategies: they determine the move of the cops solely from the current position in the game. One could define a *general strategy* to be a function that chooses the moves based on the full history of the game, i.e., a function

$$M : \left(V(G)^{[\leq k]} \times (V(G) \cup \{*\})^r\right)^{[<\omega]} \to V(G)^{[\leq k]}.$$

The tree associated with such a strategy may be inconsistent: for example, the move made in a position with two cops on the graph may depend on which of the cops was last to move. However, given the tree associated with a general winning strategy M, we can produce a winning strategy μ that uses the same number of cops and that is monotone if M is. We summarize the above observations with the following.

PROPOSITION 1. *There is a winning (k, r)-strategy μ for a graph G if and only if there is a winning general (k, r)-strategy M for G. Further, μ may be chosen to be monotone if M is monotone.*

In program 1, the moves of the cops depend only on the knowledge of which component of the tree contains the robber and not on the precise vertex he occupies. With an eye to the situation with more than one robber, we can say that the move of the cops from position S depends only on the knowledge of how many robbers are in each component of $T - S$.

In fact, the cops do not lose any strength if their information is restricted in this way. For this, given an $S \in V(G)^{[\leq k]}$ and $\mathbf{R}, \mathbf{R}' \in (V(G) \cup \{*\})^r$, we say that $\mathbf{R} \equiv_S \mathbf{R}'$ if every component of $G - S$ that contains m robbers in $\mathbf{R}$ also contains m robbers in $\mathbf{R}'$ (where $G - S$ is the graph that results from deleting the vertices in S from G). Notice that $\equiv_S$ is an equivalence relation. We call a (k, r)-strategy *smooth* if, for every $\mathbf{R}, \mathbf{R}' \in (V(G) \cup \{*\})^r$ where $\mathbf{R} \equiv_S \mathbf{R}'$, we have $\mu(S, \mathbf{R}) = \mu(S, \mathbf{R}')$. That is, the cops' moves depend only on the number of robbers in each component of $G - S$ and not on their locations within these components.

LEMMA 2. *There is a winning (k, r)-strategy in G if and only if there is a smooth winning (k, r)-strategy in G.*

Proof. Let μ be a winning (k, r)-strategy in G. For each $S \in V(G)^{[\leq k]}$, let $\mathcal{A}_S$ be the set of $\equiv_S$-equivalence classes of robber positions. For each $A \in \mathcal{A}_S$ we select an arbitrary representative $\mathbf{R}_{A,S}$. Now, define a strategy μ' by putting $\mu'(S, \mathbf{R}) = \mu(S, \mathbf{R}_{A,S})$ whenever $\mathbf{R} \equiv_S \mathbf{R}_{A,S}$.

It is clear that μ' is smooth; it remains to show that it is winning. Let $\mathcal{P}' = S'_0, \mathbf{R}'_0, S'_1, \mathbf{R}'_1, \ldots$ be any μ'-play. From the definition of μ', there is a μ-play $\mathcal{P} = S'_0, \mathbf{R}_0, S'_1, \mathbf{R}_1, \ldots$ such that, for all $i \geq 0$, $\mathbf{R}_i \equiv_{S_i} \mathbf{R}'_i$. Since the possible moves of each robber depend on his free space and not on his precise position in the graph, any move by the cops that captures a robber in $\mathcal{P}$ must also capture a robber in $\mathcal{P}'$. $\mathcal{P}$ is a winning play, so $\mathcal{P}'$, and hence μ', must also be winning. $\square$

Note that the strategies referred to when considering smoothness are not necessarily monotone but that, in the above proof, μ' is monotone if μ is.

Finally, in this section, we show that the parameters we have defined are closed under taking minors. Recall that G is a minor of H (written $G \preccurlyeq H$) if G can be constructed from H by a sequence of vertex deletions, edge deletions, and edge contractions, where an edge contraction is the deletion of two adjacent vertices u and v in H, followed by the addition of a new vertex w adjacent to all former neighbors of the deleted vertices.

PROPOSITION 3. *If $G \preccurlyeq H$, then, for any r, $\mathbf{mvams}(G,r) \leq \mathbf{mvams}(H,r)$.*

Proof. Let μ be a smooth, monotone, winning (k,r)-strategy for H, and let T_μ be the tree representing μ. We may assume that G is formed from H by deleting a single isolated vertex or deleting or contracting a single edge, since deletion of a nonisolated vertex may be achieved by first deleting all its edges.

Suppose $G = H - v$ for some $x \in V(H)$. Since x is isolated in H, any move of the cops involving x must be either a placement or a removal. Let T'_μ be the tree that results from deleting x from every vertex label in T_μ and replacing x with $*$ in every edge label. Clearly, T'_μ is a (possibly defective) tree corresponding to a monotone (k,r)-strategy for G.

Suppose $G = H - e$ for some edge $e = xy \in E(H)$. The only alterations we need to make to T_μ are to deal with slides along the now-deleted edge. Suppose $v \in V(T_\mu)$ is labeled S and sends an edge labeled $\mathbf{R}_1$ to vertex v_1, labeled S', such that $S \triangle S' = e$. Let $\mathbf{R}_1, \ldots, \mathbf{R}_\ell$ enumerate the $\equiv_S$-equivalence class of $\mathbf{R}_1$. By smoothness, v also has children $v_2, \ldots, v_\ell$ such that the edge vv_i is labeled $\mathbf{R}_i$ and v_i is labeled S' for each $i \in \{2, \ldots, \ell\}$. We may assume, without loss of generality, that $x \in S$, i.e., that the slide is from x to y. By monotonicity, the only neighbor of x in H that can be in the robbers' free space is y. Therefore, in G, no neighbor of x is in the robbers' free space, and we can replace the slide $x \to y$ with a removal from x followed by a placement to y. For each $i \in \{1, \ldots, \ell\}$, add a new vertex w_i, labeled $S - x$, and an edge vw_i, labeled $\mathbf{R}_i$. For each $j \in \{1, \ldots, \ell\}$, make a copy of the subtree of T_μ rooted at v_j, and add an edge labeled $\mathbf{R}_j$ from w_i to the root of the jth copy.

Finally, suppose G is the result of contracting the edge xy in H to give a new vertex which we denote v_{xy}. To construct a (possibly defective) strategy tree for G, it suffices to substitute v_{xy} for both x and y in all vertex and edge labels in T_μ. A robber on v_{xy} in G can reach any vertex reachable by a robber on x or y in H. The effect for the cops is as follows:

- $\mathbf{place}(x)$ becomes a null move if there was already a cop on y and, if not, becomes $\mathbf{place}(v_{xy})$;
- $\mathbf{slide}(x \to y)$ becomes a null move;
- for $z \neq y$, $\mathbf{slide}(x \to z)$ becomes $\mathbf{place}(z)$ if there is a cop on y and, if not, becomes $\mathbf{slide}(v_{xy} \to z)$;
- for $z \neq y$, $\mathbf{slide}(z \to x)$ becomes $\mathbf{remove}(z)$ if there is a cop on y and, if not, becomes $\mathbf{slide}(z \to v_{xy})$;
- $\mathbf{remove}(x)$ becomes a null move if there is a cop on y and, if not, becomes $\mathbf{remove}(v_{xy})$.

The cases for moves involving y are symmetric. ∎

3. Variants of monotonicity. In the previous section, we defined the concept of monotonicity for plays and strategies. These definitions are natural extensions of the case with only one robber but are not the only ones. In this section, we consider two further natural definitions of monotonicity, which turn out to be equivalent to our first definition, and we begin an investigation of the cost of monotonicity.

Let $\mathcal{P} = S_0, \mathbf{R}_0, S_1, \mathbf{R}_1, \ldots$ be a μ-play. We say that $\mathcal{P}$ is *pointwise monotone* if, for each $j \in \{1, \ldots, r\}$ and each $i \geq 0$, $F^j_{i+1} \subseteq F^j_i$; i.e., no single robber's set of free positions ever increases. Also, we say that $\mathcal{P}$ is *cop-monotone* if, for each $v \in V(G)$, the set

$$s_{\mathcal{P}}(v) = \{i \mid v \in S_i \text{ and } V(\mathbf{R}_i) \neq \emptyset\}$$

is an interval of $\mathbb{N}$—that is, once the cops have left a vertex, they never return to it as long as there are robbers in the graph. Observe that any cop-monotone μ-play must be a winning μ-play because plays are infinite and G is not, so the cops must eventually revisit a vertex if the robbers live forever. We say that a (k, r)-strategy μ is monotone according to one of the above definitions if all μ-plays are.

LEMMA 4. *Let G be a graph, and let k and r be positive integers. The following are equivalent:*

1. *there is a monotone winning (k, r)-strategy in G;*
2. *there is a pointwise-monotone winning (k, r)-strategy in G;*
3. *there is a cop-monotone (k, r)-strategy in G.*

Proof. $(2) \Rightarrow (1)$ follows trivially from the definitions.

$(3) \Rightarrow (2)$. Let μ be a cop-monotone strategy, and suppose that, for some μ-play $\mathcal{P} = S_0, \mathbf{R}_0, S_1, \mathbf{R}_1, \ldots$, there is a step, say, from S_i to S_{i+1}, where the free space of the jth robber ($1 \leq j \leq r$) increases, i.e., that $F^j_{i+1} \supset F^j_i$. It follows that there must have been a cop removed or slid from some vertex $v \in \partial_G(F^j_i)$, the set of vertices in $V(G) - F^j_i$ that are adjacent to at least one vertex in F^j_i. There is a μ-play $\mathcal{P}'$ that follows $\mathcal{P}$ until S_{i+1} and in which the jth robber moves to the newly vacated vertex v and stays there at all subsequent moves. But now, this robber cannot be caught unless v is revisited, contradicting the assumed cop-monotonicity of μ.

For $(1) \Rightarrow (3)$, the idea is that the cops never need to visit a vertex that is not in the robbers' free space because such a move can never decrease the free space. Therefore, the move does not contribute to the capture of the robbers and can safely be omitted. Thus, every move made by the cops may be assumed to be a removal, a placement, or a slide into the robbers' free space and, since the free space is monotonically decreasing, each such move must be the first time the target vertex has been visited.

Formally, let μ be a monotone winning strategy which we may assume, by Lemma 2, to be smooth. Let $\mathcal{E}$ be the set of the essential parts of all μ-plays. For any μ-play $\mathcal{P} \in \mathcal{E}$, let $c(\mathcal{P})$ be the number of vertices revisited by the cops (i.e., the number of vertices v for which $s_{\mathcal{P}}(v)$ is not an interval), and let $c(\mu) = \sum_{\mathcal{P} \in \mathcal{E}} c(\mathcal{P})$. $c(\mu)$ is well defined, as $\mathcal{E}$ is finite; further, $c(\mu) = 0$ if and only if μ is cop-monotone.

Suppose μ is not cop-monotone. We construct a (k, r)-strategy μ' with $c(\mu') < c(\mu)$. Repeated applications of this transformation will yield the desired cop-monotone strategy.

Let $\mathcal{P} = S_0, \mathbf{R}_0, S_1, \mathbf{R}_1, \ldots$ be a non–cop-monotone μ-play, and let v_{i+1} be a vertex that is revisited for the first time in the step from S_i to S_{i+1}. This means that $S_{i+1} - S_i = \{v_{i+1}\}$—the move is a placement or a slide. Let H be the union of the connected components of $G - S_i$ that intersect $V(\mathbf{R}_i)$, and let $S^* = \partial_G H$. Notice that $S^* \subseteq S_i$ since, otherwise, $\mathcal{P}$ is not monotone.

We cannot have $v_{i+1} \in V(H)$ since, by monotonicity, none of the vertices in H has yet been visited. We cannot have $v_{i+1} \in S^*$ as there are already cops on every vertex of S^*, so these vertices cannot be the target of a placement or a slide. Further, if the move is a slide, from the single vertex $v_i \in S_i - S_{i+1}$, then $v_i \notin S^*$: suppose $v_i \in S^*$; since $v_{i+1} \notin H$, v_i becomes part of the robbers' free space, contradicting the monotonicity of μ.

Let $T_0 = S_i$ and, for $j \geq 0$, let $T_{j+1} = \mu(T_j, \mathbf{R}_i)$. Let h be minimal such that $T_{h+1} \cap V(H) \neq \emptyset$. Thus, $T_1 = S_{i+1}$ and T_h is the first move at which the cops will play into the robbers' free space if the robbers stand still. Notice that, in any μ-play that includes the position $S_i, \mathbf{R}_i$, the positions $T_1, \ldots, T_h$ will follow because, by smoothness, the moves of the cops depend only on the free space of the robbers, which does not change during the quoted sequence (indeed, this remains true if we replace $\mathbf{R}_i$ with any $\mathbf{R} \equiv_{S_i} \mathbf{R}_i$). As before, the monotonicity of μ implies that $S^* \subseteq T_j$ for $j \in \{0, \ldots, h\}$.

We now define μ'. The idea is to replace the sequence $T_0, \ldots, T_h$ with a new sequence of moves that performs the least number of removals and placements to move the cops from T_0 to T_h but omits the move to v_{i+1}. Since none of these moves is in H, the robbers' free space remains the same and monotonicity is preserved. Toward this end, let $T_0 - T_h = \{x_1, \ldots, x_p\}$ and $T_h - T_0 - \{v_{i+1}\} = \{y_1, \ldots, y_q\}$. For any $\mathbf{R} \equiv_{S_i} \mathbf{R}_i$, set

$$\begin{aligned}
\mu'(T_0, \mathbf{R}) &= T_0 - \{x_1\}, \\
\mu'(T_0 - \{x_1\}, \mathbf{R}) &= T_0 - \{x_1, x_2\} \\
&\vdots \\
\mu'(T_0 - \{x_1, \ldots, x_{p-1}\}, \mathbf{R}) &= T_0 - \{x_1, \ldots, x_p\}\,.
\end{aligned}$$

Writing T' for $T_0 - \{x_1, \ldots, x_p\}$, set

$$\begin{aligned}
\mu'(T', \mathbf{R}) &= T' \cup \{y_1\}, \\
\mu'(T' \cup \{y_1\}, \mathbf{R}) &= T' \cup \{y_1, y_2\} \\
&\vdots \\
\mu'(T' \cup \{y_1, \ldots, y_{q-1}\}, \mathbf{R}) &= T' \cup \{y_1, \ldots, y_q\}\,.
\end{aligned}$$

Note that $T' \cup \{y_1, \ldots, y_q\} = T_h - \{v_{i+1}\}$, and observe that the placements and removals defined above do not involve placement to the vertex v_{i+1}. Also, any vertex that is revisited in the new chain of moves would have been revisited anyway if the old chain of moves had been made. However, so far, the new chain does not revisit v_{i+1}, which was revisited in the old chain. To guarantee that v_{i+1} is not revisited in any future sequence of moves, we set $\mu'(S - \{v_{i+1}\}, \mathbf{R}) = \mu(S, \mathbf{R}) - \{v_{i+1}\}$ for any $S \subseteq V(H) \cup T_h$ and any $\mathbf{R}$ where $V(\mathbf{R}) \subseteq V(H)$. Otherwise, put $\mu'(S, \mathbf{R}) = \mu(S, \mathbf{R})$. We now have $c(\mu') < c(\mu)$, as required. ☐

We have shown the natural definitions of monotonicity to be equivalent, but is monotonicity important? Suppose we have r robbers in a tree T. We can modify program 1 so that, instead of letting T' be any component containing a robber, we set T' to be the component containing the ith robber, where i is minimal among those robbers who have not yet been caught; see program 2. This gives a program that catches the first robber (and any other robbers foolish enough to follow him), then

Search program 2. $\Pi(T, r)$ to capture r robbers in a tree T.

place(v), where v is any vertex of T.
Let $\mathbf{R} = [u_1 \dots u_r] \leftarrow$ *robbers_positions()*.
Let $T' \leftarrow T$.
While $V(\mathbf{R}) \cap V(T') \neq \emptyset$,
  Let i be minimal such that $u_i \neq *$.
  Let T' be the connected component of $T - v$ containing u_i,
    and let w be the (unique) vertex of T' adjacent to v.
  slide$(v \to w)$.
  Let $v \leftarrow w$.
  Let $\mathbf{R} \leftarrow$ *robbers_positions()*.
remove(v).

the second, and so on. There are two things to notice about this program: first, it is not monotone; second, it is winning against any number of robbers, without needing any more cops.

The same technique can be applied to transform any search program for one robber (on an arbitrary collection of graphs) into a nonmonotone program for any number of robbers. On the other hand, it is clear that monotonically searching for $r > 1$ robbers requires at least as many cops as does monotonically searching for a single robber. We summarize these observations in the following lemma.

LEMMA 5. *For any graph G and positive integer r,*

$$\mathbf{vams}(G, r) = \mathbf{vams}(G, 1),$$
$$\mathbf{mvams}(G, r) \geq \mathbf{mvams}(G, 1)\,.$$

Thus, allowing nonmonotone strategies may make it easier to search for many robbers. This raises the question of what the cost of requiring monotonicity is when facing a crime wave. Given a graph G and $r \geq 1$, what is the ratio below?

$$\frac{\mathbf{mvams}(G, r)}{\mathbf{mvams}(G, 1)}.$$

In sections 5 and 6, we give a full answer for trees and an upper bound for general graphs. We postpone this until we have established some necessary results in the next section.

4. Invisible robbers. In this section we give brief descriptions of two game variants where the robbers are invisible and the cops must, therefore, determine their moves without reference to the robbers' position. In one of the variants we consider, the robber is active; in the other, he is lazy. Recall that an active robber can move at each round of the game, but a lazy robber may move only when a cop moves onto the vertex he occupies.

In both cases, as the robbers are now invisible, the game is no longer interactive and the cops' moves may be given in advance as a "predetermined" strategy. Thus, we define a *k-strategy* for k cops to be any consistent sequence $\mathcal{S} = S_0, S_1, \dots$ of sets in $V(G)^{[\leq k]}$.

Given such a strategy, we define the free space of an invisible, active robber to be the sequence

$$F_0 = V(G),$$
$$F_{i+1} = \{y \in V(G) - S_{i+1} \mid \text{there is an } (S_i, S_{i+1})\text{-avoiding } (x,y)\text{-path for some } x \in F_i\}\,.$$

We say that $\mathcal{S}$ is *monotone* if $F_{i+1} \subseteq F_i$ for all $i \geq 0$ and $\mathcal{S}$ is *winning* if $F_i = \emptyset$ for some $i \geq 1$. Since the game is not interactive, we do not explicitly define plays, which will not feature in our analysis.

The *nonmonotone* and *monotone invisible active mixed search number* of a graph G are defined as follows:

$$\mathbf{iams}(G) = \min\{k \mid \text{there exists a winning } k\text{-strategy on } G\},$$
$$\mathbf{miams}(G) = \min\{k \mid \text{there exists a monotone winning } k\text{-strategy on } G\}.$$

It is known that $\mathbf{iams}(G) = \mathbf{miams}(G)$ [3]; that is, when searching for an invisible, active robber, insisting that the cops win the game monotonically does not increase the number of cops required. We also observe that searching for an active, invisible robber in a n-vertex graph G is equivalent to searching for n visible, active robbers. Intuitively, an invisible robber could be anywhere within his free space, while, with n robbers, there are plays in which every vertex of the free space really does contain a robber.

LEMMA 6. *For any graph G of order n,* $\mathbf{miams}(G) = \mathbf{mvams}(G, n)$.

Proof. We prove first that $\mathbf{miams}(G) \leq \mathbf{mvams}(G, n)$. Suppose we have a monotone winning (k, n)-strategy μ for the cops. Consider a μ-play $\mathcal{P} = S_0, \mathbf{R}_0, S_1, \mathbf{R}_1, \ldots$ where, for all $i \geq 0$, $V(\mathbf{R}_i) = F_i$. Such a play exists, since $V(R_0) = F_0 = V(G)$ and, by definition of F_{i+1}, it is possible for the robbers occupying the vertices F_i to move to the vertices in F_{i+1}. Notice, now, that the sequence $S_0, S_1, \ldots$ is a winning monotone k-strategy against an invisible, active robber.

For the converse, let $\mathcal{S} = S_0, S_1, \ldots$ be a monotone winning k-strategy against one invisible, active robber. We define a (k, n)-strategy μ by putting $\mu(S_i, \mathbf{R}) = S_{i+1}$ for any $i \geq 0$ and any $\mathbf{R} \in (V(G) \cup \{*\})^n$ with $V(\mathbf{R}) \subseteq F_{i+1}$. Any μ-play is monotone and winning because, no matter what moves the robbers make, $V(\mathbf{R}_i) \subseteq F_i$ for all i and the sequence $F_0, F_1, \ldots$ diminishes monotonically to the empty set. ∎

The case of an invisible, lazy robber is similar to the active case but with the difference that now, if the cops are at S and the robber at vertex v, then, when the cops move to S', the robber must stay at v unless $v \in S'$, in which case he can move along any (S, S')-avoiding path in the graph, as before. Thus, the robber moves only when a cop lands on his vertex.

We define free space, k-strategies, monotonicity, and winning against a lazy, invisible robber in the same way as in the active case and write $\mathbf{milms}(G)$ and $\mathbf{ilms}(G)$ for the corresponding *monotone* and *nonmonotone, invisible, lazy, mixed search number*, respectively.

LEMMA 7. *For any graph G,* $\mathbf{milms}(G) = \mathbf{mvams}(G, 1)$.

Proof. We prove first that $\mathbf{milms}(G) \leq \mathbf{mvams}(G, 1)$. Suppose we have a monotone winning $(k, 1)$-strategy μ for G, and let $\mathcal{E}$ be the set of the essential parts of all μ-plays. We can construct a monotone winning k-strategy against one invisible, lazy robber by taking an arbitrary concatenation of all of the sequences in $\mathcal{E}$.

We now show that $\mathbf{mvams}(G, 1) \leq \mathbf{milms}(G)$. Let $\mathcal{S} = S_0, S_1, \ldots$ be a monotone winning k-strategy against one invisible, lazy robber. We will describe a search program against a visible, active robber. The first move is to place a cop on the vertex in S_1. Suppose that, at some stage, the cops occupy the vertices of some set S (not necessarily a set in $\mathcal{S}$). Let H be the connected component of $G - S$ that contains the robber, and let $S^* = \partial_G H$.

Now, let i be minimal such that S_i contains a vertex in H. We remove any cops that may be in $S - S_{i-1}$ and then play the move m that transforms S_{i-1} to S_i which, by definition of i, is not a removal. It is clear that we can play this move if it is a placement. If it is a slide, it must, by construction, be from a vertex in S^*. S must contain S^* by definition and, since $\mathcal{S}$ is a monotone strategy, S_{i-1} must also contain S^*: m is the first attack on H and the robber would be able to escape from that component if its boundary were not guarded.

This establishes that we have a monotone strategy. To see that it is winning, observe that, at each step, the robber's free space is decreased by at least one vertex (the target of m) so must, eventually, become empty. ∎

We define the *proper pathwidth* of a graph G to be $\mathbf{ppw}(G)$, the least k for which $G \preccurlyeq K_k \times P$ for some path P. (That is, $G \preccurlyeq G'$ for the graph G' formed from P by replacing the vertices with disjoint copies of K_k and adding a matching between the vertices of cliques corresponding to vertices adjacent in P.) Similarly, define the *proper treewidth* of a graph G as $\mathbf{ptw}(G)$, the least k for which $G \preccurlyeq K_k \times T$ for some tree T. It can be shown that $\mathbf{miams}(G) = \mathbf{ppw}(G)$ and $\mathbf{milms}(G) = \mathbf{ptw}(G)$ (see, e.g., [9, 12]).

COROLLARY 8. *For any graph G and any positive integer r,*

$$\mathbf{ptw}(G) \leq \mathbf{mvams}(G, r) \leq \mathbf{ppw}(G).$$

Proof. The proof is immediate from Lemmas 6 and 7 and the observation that, for any graph G of order n and any $r > n$, $\mathbf{mvams}(G, r) = \mathbf{mvams}(G, n)$, since the robbers can never occupy more than n distinct vertices. ∎

We could also consider multiple invisible robbers. We will not define the relevant games formally, but the informal remarks that follow should convince the reader that it would not be worth the effort to do so.

In the case of invisible, active robbers, it is clear that $\mathbf{miams}(G, r) = \mathbf{miams}(G)$ for any $r \geq 1$; essentially, no graph parameter defined through mixed search can ever be bigger than proper pathwidth. For invisible, lazy robbers, we must consider the conditions under which the robbers may move. The simplest scenario is that each robber may move only when a cop lands on the vertex he occupies. Define the *nonmonotone* and *monotone, invisible, lazy mixed search number* of a graph G to be, respectively, $\mathbf{ilms}(G, r)$ and $\mathbf{milms}(G, r)$, the least k such that there is a winning (respectively, monotone winning) k-strategy against r invisible, lazy robbers. In this case, it is not hard to see that $\mathbf{ilms}(G, r) = \mathbf{ilms}(G, 1)$ (because, as usual, we can iterate the strategy for one robber to catch r robbers) and $\mathbf{milms}(G, r) = \mathbf{milms}(G, 1)$ (the strategy used to prove Lemma 7 works as well for $r \geq 1$ lazy robbers as for one).

On the other hand, suppose that, when a cop lands on a vertex occupied by any robber, this fact is communicated to all the robbers, who may all move. Define the *nonmonotone* and *monotone, invisible, communicating, lazy mixed search numbers* of a graph G to be, respectively, $\mathbf{iclms}(G, r)$ and $\mathbf{miclms}(G, r)$, the least k such that there is a winning (respectively, monotone winning) k-strategy against r invisible,

Search program 3. $\Pi(T, v, r)$ to capture r robbers in a tree T monotonically.

place(v)
Let $\mathbf{R} \leftarrow$ *robbers_positions()*.
While $V(\mathbf{R}) \neq \emptyset$,
    Let $T_1, \ldots, T_\ell$ be the connected components of $T - v$
        containing at least one and at most $\lfloor \frac{r}{2} \rfloor$ robbers.
    For $i \in \{1, \ldots, \ell\}$,
        Choose any vertex $v_i \in V(T_i)$.
        Let r_i be the number of robbers in T_i.
        Call $\Pi(T_i, v_i, r_i)$.
    Let $\mathbf{R} \leftarrow$ *robbers_positions()*.
    if $V(\mathbf{R}) \cap V(T) \neq \emptyset$ (i.e., robbers remain in T), then
        Let T' be the unique connected component of $T - v$ where
            $V(\mathbf{R}) \subseteq V(T')$, and let w be the vertex of T'
            adjacent to v in T.
        slide$(v \rightarrow w)$.
        Let $v \leftarrow w$ and let $T \leftarrow T'$.
remove(v).

communicating, lazy robbers. We still have $\mathbf{iclms}(G, r) = \mathbf{ilms}(G)$ and, with just one robber, of course, $\mathbf{miclms}(G, 1) = \mathbf{milms}(G) = \mathbf{ptw}(G)$ since a single robber has nobody to communicate with. However, for any $r \geq 2$, having r invisible, communicating, lazy robbers is as bad as having an active, invisible robber: essentially, whenever the cops move to a vertex in the robbers' free space, they may disturb a robber, and, if they do, all the robbers may move. Thus, after any move, the cops must ensure that the entire boundary of the robbers' free space is guarded, just as in the active, invisible case. Hence, for $r \geq 2$, $\mathbf{miclms}(G, r) = \mathbf{miams}(G) = \mathbf{ppw}(G)$. We summarize these observations in the following theorem.

THEOREM 9. *For any graph G and any integers $r \geq 1$ and $s \geq 2$,*

$$\begin{aligned}
\mathbf{ilms}(G, r) &= \mathbf{iclms}(G, r) = \mathbf{ilms}(G), \\
\mathbf{milms}(G, r) &= \mathbf{miclms}(G, 1) = \mathbf{milms}(G) = \mathbf{ptw}(G), \\
\mathbf{miclms}(G, s) &= \mathbf{miams}(G, r) = \mathbf{miams}(G) = \mathbf{ppw}(G)\,.
\end{aligned}$$

We also note that it is believed but not yet proven[1] that $\mathbf{ilms}(G) = \mathbf{milms}(G)$.

5. Upper bounds. In this section, we demonstrate upper bounds for the value of $\mathbf{mvams}(G, r)$ for trees, in particular, and for all graphs.

LEMMA 10. *If T is a tree, then* $\mathbf{mvams}(T, r) \leq \lfloor \log r \rfloor + 1$.

Proof. Let $\Pi(T, r)$ be the search program that calls program 3 with v assigned to be any vertex of T.

We must prove that $\Pi(T, r)$ is winning and monotone and uses at most $\lfloor \log r \rfloor + 1$ cops. For this we use induction on the logarithm of the number of robbers. For the base case, notice that $\Pi(T, r)$ degenerates to program 1 when $r = 1$, as the program operates exclusively in the single component of $T - v$ containing $\lceil \frac{r}{2} \rceil = 1$ robber.

[1] A flawed proof appeared in [12].

Suppose that $\Pi(T, \lfloor \frac{r}{2} \rfloor)$ defines a winning, monotone $(q, \lfloor \frac{r}{2} \rfloor)$-strategy, where $q = \lfloor \log \frac{r}{2} \rfloor + 1 = \lfloor \log r \rfloor$. We now show that $\Pi(T, r)$ defines a winning, monotone $(q+1, r)$-strategy.

Before each slide move to w, each component of $T - v$ except for the one containing w contained at most $\lfloor \frac{r}{2} \rfloor$ robbers and has, by the inductive hypothesis, already been searched monotonically. Therefore, after each slide move, the free positions of the robbers have been updated from $V(T)$ to $V(T')$, where $V(T') \subset V(T)$. As the free positions for the robbers diminish, the program is monotone; and, as they diminish properly, the program is winning.

By the inductive hypothesis, each call to $\Pi(T_i, v_i, r_i)$ requires q cops. Meanwhile, there is only one additional cop in T (the cop on v), so $\Pi(T, r)$ uses $q + 1$ cops, as required. $\square$

The following upper bound on $\mathbf{mvams}(T, r)$ is immediate from the previous lemma and Corollary 8.

COROLLARY 11. *For any tree T and for any positive integer r,*

$$\mathbf{mvams}(T, r) \leq \min \{\mathbf{ppw}(G), \lfloor \log r \rfloor + 1\}.$$

Our bound for trees leads to a bound for general graphs, obtained by considering tree decompositions.

THEOREM 12. *For any graph G and any positive integer r,*

$$\mathbf{mvams}(G, r) \leq \min \{\mathbf{ppw}(G), \mathbf{ptw}(G) \cdot (\lfloor \log r \rfloor + 1)\}.$$

Proof. Let $q = \lfloor \log r \rfloor + 1$.

By Corollary 8, it is enough to show that $\mathbf{mvams}(G, r) \leq \mathbf{ptw}(G) \cdot q$. Assuming that $\mathbf{ptw}(G) \leq k$, we have $G \preccurlyeq G' = K_k \times T$ for some tree T. We assume that the vertices of the clique in G' corresponding to the vertex $v \in T$ are $K(v) = \{v_1, \ldots, v_k\}$ and, for every edge $uv \in T$, the corresponding edges in G' are $u_1v_1, \ldots, u_kv_k$. For each $S \subseteq V(T)$, let $K(S) = \bigcup_{v \in S} K(v)$.

By Lemma 10, $\mathbf{mvams}(T, r) \leq q$, so there is a monotone winning (q, r)-strategy μ for T. We use μ to construct a monotone, winning (kq, r)-strategy μ' for G'. The idea is that we simulate a single cop on $v \in T$ with k cops, one on each vertex of $K(v) \subseteq G'$. Each placement, removal and slide is replaced by the equivalent operation on each of these k cops in turn.

Formally, let $S \in V(T)^{[\leq q]}$, let $\mathbf{R} \in (V(T) \cup \{*\})^r$, and let $S' = \mu(S, \mathbf{R})$. There are three cases, depending on the type of the move from S to S'.

Placement. Let $\{v\} = S' - S$. For any $j \in \{1, \ldots, k\}$, let $S_j = K(S) \cup \{v_1, \ldots, v_{j-1}\}$. For any $\mathbf{R}'$ with $V(\mathbf{R}') \subseteq K(V(\mathbf{R})) - \{v_1, \ldots, v_{j-1}\}$, we set $\mu'(S_j, \mathbf{R}') = S_j \cup \{v_j\}$.

Removal. Let $\{v\} = S - S'$. For any $j \in \{1, \ldots, k\}$, let $S_j = K(S) - \{v_1, \ldots, v_{j-1}\}$. For any $\mathbf{R}'$ with $V(\mathbf{R}') \subseteq K(V(\mathbf{R})) \cup \{v_1, \ldots, v_{j-1}\}$, we set $\mu'(S_j, \mathbf{R}') = S_j - \{v_j\}$.

Sliding. Let $\{v\} = S - S'$ and $\{w\} = S' - S$. For any $j \in \{1, \ldots, k\}$, let $S_j = \big(K(S) - \{v_1, \ldots, v_{j-1}\}\big) \cup \{w_1, \ldots, w_{j-1}\}$. For any $\mathbf{R}'$ with $V(\mathbf{R}') \subseteq \big(K(V(\mathbf{R})) \cup \{v_1, \ldots, v_{j-1}\}\big) - \{w_1, \ldots, w_{j-1}\}$, we set $\mu'(S_j, \mathbf{R}') = (S_j - \{v_j\}) \cup \{w_j\}$.

Notice that the fact that μ is winning and monotone implies the same for μ'. Moreover, in each μ'-play any set $S \in V(T)^{[\leq q]}$ corresponds to a sequence of k sets in $V(G')^{[\leq kq]}$. Therefore, $\mathbf{mvams}(G', r) \leq kq$ and thus $\mathbf{mvams}(G, r) \leq kq$, by Proposition 3. $\square$

6. Lower bounds. We now give lower bounds for $\mathbf{mvams}(T, r)$ for trees T. We introduce a general form of graph composition and analyze the search numbers of graphs formed by such compositions. We define the composition for general graphs, though our main use of the construction will be for trees.

We say that graphs $G_0, \ldots, G_3$ are *k-connectable* if

- for $0 \leq i \leq 3$, G_i is k-connected;
- G_1, G_2, and G_3 are pairwise disjoint;
- for $1 \leq i \leq 3$, $|U_i| = k$, where $U_i = V(G_0) \cap V(G_i)$; and
- for $1 \leq i < j \leq 3$ and $h \in \{1, 2, 3\} - \{i, j\}$, $G_0 - U_h$ contains a set P_{ij} of k pairwise vertex-disjoint paths from U_i to U_j.

Note that the k-connectedness of G_0 already implies the existence in that graph of a set of k pairwise vertex-disjoint paths from U_i to U_j, but these do not necessarily avoid U_h.

Let G be a graph, with $x \in V(G)$ and $U \subseteq V(G) - \{x\}$. An *(x, U)-fan* is a set of paths in G, one from x to each vertex in U, where the paths are pairwise disjoint, except for the common endpoint x.

LEMMA 13. *Let $G_0, \ldots, G_3$ be k-connectable. Then $G = G_0 \cup \cdots \cup G_3$ is k-connected.*

Proof. It suffices to show that, for any $u, v \in V(G)$, G contains k independent (u, v)-paths. If $u, v \in V(G_i)$ for some i, then the result follows immediately from the k-connectedness of G_i. So, suppose that there are $i < j$ such that $u \in V(G_i) - V(G_j)$ and $v \in V(G_j) - V(G_j)$. There are two cases.

If $i = 0$, then, by [4, Theorem 2.6], G_0 contains a (u, U_j) fan and G_j contains a (v, U_j) fan. Since $V(G_0) \cap V(G_j) = U_j$, these fans are disjoint, except for the vertices in U_j. Their union is, therefore, a set of k independent (u, v)-paths in G.

If $i > 0$ then, as above, G_i contains a (u, U_i)-fan and G_j contains a (u, U_j)-fan and these fans are disjoint. The union of the two fans and the paths P_{ij} is a set of k independent (u, v)-paths in G. ☐

The following is the key technical result of this section.

LEMMA 14. *Let $G_0, \ldots, G_3$ be k-connectable, with $\mathbf{mvams}(G_i, \lfloor \frac{r}{2} \rfloor) \geq q$ for each i, and let $G = G_0 \cup \cdots \cup G_3$. Then $\mathbf{mvams}(G, r) \geq q + k$.*

Proof. Suppose, toward a contradiction, that $\mathbf{mvams}(G, r) < q + k$, and let μ be a smooth, monotone, winning $(q + k - 1, r)$-strategy for G. By Lemma 13, G is k-connected. Hence, whenever there are robbers in the graph and the position of the cops is S, with $|S| < k$, the robbers' free space is the whole of $G - S$. Let $\mathcal{P} = S_0, \mathbf{R}_0, S_1, \mathbf{R}_1, \ldots$ be a play, and let α be minimal such that

- the free space of some robber in $\mathbf{R}_\alpha$ does not include the whole of $V(G_i) - S_\alpha$ for some $i \geq 1$, or
- $|V(G_i) \cap S_\alpha| \geq k$ for some $i \geq 1$.

α is well defined because μ is winning. We may assume that the robbers, who are aware of the cops' strategy μ, arrange to maximize α. By smoothness, the cops make the same moves in all plays up to (and including) the αth move, as long as the robbers behave as we have described.

As argued above, $|S_\alpha| \geq k$; otherwise, the free space of every robber is just $V(G) - S_\alpha$ and no G_i contains k cops. We may assume, without loss of generality, that $|V(G_1) \cap S_\alpha| < k$ and $|V(G_2) \cap S_\alpha| < k$. (If necessary, rename the parts to achieve this.) Let $S'_\alpha = S_\alpha - (V(G_1) \cap V(G_2))$. By construction, $|S'_\alpha| \geq k$.

At the point when the cops make move α, the free space of every robber includes all of $V(G_1) - S_{\alpha-1}$ and all of $V(G_2) - S_{\alpha-1}$. Therefore, we may assume that $\mathbf{R}_\alpha$ has

$\lceil \frac{r}{2} \rceil$ robbers in G_1 and $\lfloor \frac{r}{2} \rfloor$ robbers in G_2. We will show that the assumption that μ is monotone and winning contradicts the hypothesis that $\mathbf{mvams}(G_1, \lfloor \frac{r}{2} \rfloor) \geq q$ and $\mathbf{mvams}(G_2, \lfloor \frac{r}{2} \rfloor) \geq q$.

Let T_μ be the labeled tree representing the strategy μ. We may delete from T_μ any subtree whose root has an incoming edge labeled $\mathbf{R}$ for some $\mathbf{R}$ containing more than $\lceil \frac{r}{2} \rceil$ robbers in G_1 or more than $\lfloor \frac{r}{2} \rfloor$ robbers in G_2. This restriction on the robbers' position can only make the game easier for the cops.

First, let T_1 be the tree formed from T_μ by deleting every vertex outside $V(G_1)$ from every vertex label and replacing every vertex outside $V(G_1)$ with $*$ in every edge label. These deletions from the labels may result in two adjacent vertices u and v having identical labels (because the corresponding move in T_μ was outside G_1) and two edges from the same vertex having identical labels (because a move within G_1 in T_μ depended on the positions of robbers outside G_1). As such, T_1 is nondeterministic, in the sense discussed in section 2.

Call the subtree of T_1 rooted at vertex x *bad* if either the label of x is a set of size at least q or there is some $\mathbf{R}$ such that, whenever the edge (x, y) is labeled $\mathbf{R}$, the subtree rooted at y is bad. Call a subtree *good* if it is not bad. We claim that T_1 is, itself, bad. Suppose not. Delete all bad subtrees from T_1 to give T_1'. Because every vertex of the resulting tree has at least one child for each possible $\mathbf{R}$, T_1' defines a winning $(q-1, \lceil \frac{r}{2} \rceil)$-strategy for G_1, contradicting the hypothesis that $\mathbf{mvams}(G_1, \lceil \frac{r}{2} \rceil) \geq \mathbf{mvams}(G_1, \lfloor \frac{r}{2} \rfloor) \geq q$. (In fact, the strategy defined by T_1' may be nondeterministic, but we may take an arbitrary deterministic restriction.)

Now, let T_1^* be the tree that results from deleting all good subtrees from T_1. T_1^* is a tree of all plays where the robbers force there to be q cops in G_1 and can be seen as a certificate of the fact that no monotone winning $(q+k-1, r)$-strategy for G can induce a monotone winning $(q-1, \lceil \frac{r}{2} \rceil)$-strategy for G_1.

Let T be the subtree of T_μ consisting of those vertices in T_1^*, and let T_2 be the tree made from T by deleting every vertex outside $V(G_2)$ from every vertex label and replacing every vertex outside $V(G_2)$ with $*$ in every edge label. T_2 defines a nondeterministic strategy for G_2 in the same sense that T_1 does for G_1. Because T_1 is a strategy for G_1, it includes responses for every possible position $\mathbf{R}$ of the robbers within that subgraph. In turn, every position of robbers in G whose restriction to G_1 is $\mathbf{R}$ will produce the same response within T_1': in particular, then, T_2 contains a vertex corresponding to this position.

Using the hypothesis that $\mathbf{mvams}(G_2, \lfloor \frac{r}{2} \rfloor) \geq q$ and the same argument as for G_1, we see that T_2 is also bad. Again, define T_2^* by deleting all good subtrees from T_2. By construction, any path in T_2^* corresponds to a path in T_1^* and a path in T_μ. Choose any such path, and let $\mathcal{P} = S_0, \mathbf{R}_0, S_1, \mathbf{R}_2, \ldots$ be the corresponding μ-play. For $i \in \{1, 2\}$, let $\mathcal{P}^i = S_0^i, \mathbf{R}_0^i, S_1^i, \mathbf{R}_2^i, \ldots$ be the labels of the corresponding path in T_i^*. Note that $S_j^1 \cup S_j^2 \subseteq S_j$ and $V(\mathbf{R}_j^1) \cup V(\mathbf{R}_j^2) \subseteq V(\mathbf{R}_j)$ for all $j \geq 0$.

For $i \in \{1, 2\}$, let c_i be minimal such that $V(\mathbf{R}_{c_i}^i) = \emptyset$. Since no move of the cops can simultaneously capture robbers in both G_1 and G_2, we must have $c_1 \neq c_2$. Without loss of generality, we may assume $c_1 < c_2$. Let h be minimal such that $|S_h^1| \geq q$. Since μ is a $(q+k-1, r)$-strategy, we must have $|S_h^2| < k$, and, further, at least one of the cops originally placed on $v \in S_\alpha'$ must have been removed. This contradicts the monotonicity of μ because, after move α of the game, no robber in G_2 could reach vertex v, but now they all can. $\square$

A 3-*star composition* of disjoint, connected graphs G_1, G_2, and G_3 is any graph $\mathsf{Y}(G_1, G_2, G_3)$ formed by adding a new vertex v to $G_1 \cup G_2 \cup G_3$ and adding one edge from v to each of the three component graphs. Observe that the 3-star composition of G_1, G_2, and G_3 is a special case of the graphs $K_{1,3}$, G_1, G_2, and G_3 being 1-connectable. Hence, the following is an immediate corollary of Lemma 14.

COROLLARY 15. *Let* $G = \mathsf{Y}(G_1, G_2, G_3)$*, where, for each* $i \in \{1, 2, 3\}$*, it holds that* $\mathbf{mvams}(G_i, \lfloor \frac{r}{2} \rfloor) \geq q$*. Then,* $\mathbf{mvams}(G, r) > q$*.*

We are now ready to show that the upper bound of Corollary 11 is, in fact, an exact characterization of $\mathbf{mvams}(T, r)$ for all trees T and natural numbers r.

THEOREM 16. *For any tree* T *and* $r \geq 1$*,*

$$\mathbf{mvams}(T, r) = \min \{\mathbf{ppw}(T), \lfloor \log r \rfloor + 1\} .$$

Proof. By Corollary 11, it suffices to show that $\mathbf{mvams}(T, r) \geq \min \{\mathbf{ppw}(T), \lfloor \log r \rfloor + 1\}$. For this, we use induction on $q = \lfloor \log r \rfloor + 1$. For the base case, $q = 1$, program 1 shows that $\mathbf{mvams}(T, r) = 1 = q$. Suppose the result holds for all values smaller than q, and let T be a tree. If $\mathbf{ppw}(T) = 1$, then T is a path and $\mathbf{mvams}(T, r) = 1$ for any r, as required. Otherwise, it is known from [14] that we can write $T = \mathsf{Y}(T_1, T_2, T_3)$, where, for each i, $\mathbf{ppw}(T_i) = \mathbf{ppw}(T) - 1$. By the inductive hypothesis, $\mathbf{mvams}(T_i, \lfloor \frac{r}{2} \rfloor) \geq \min \{\mathbf{ppw}(T) - 1, q - 1\}$. By Corollary 15, $\mathbf{mvams}(T, r) \geq \min \{\mathbf{ppw}(T) - 1, q - 1\} + 1 = \min \{\mathbf{ppw}(T), q\}$, as required. ∎

We do not have a lower bound for $\mathbf{mvams}(G, r)$ for general graphs. However, we are able to demonstrate that the upper bound of Theorem 12 is reached by an infinite class of graphs. Toward this end, define the parameterized graph class $\mathcal{O}_w$ recursively as follows: $\mathcal{O}_0 = \{K_1\}$, and $G \in \mathcal{O}_{w+1}$ if and only if G is a 3-star composition of three graphs in $\mathcal{O}_w$. From [13], $\mathcal{O}_w$ contains all minor-minimal trees with proper pathwidth at least w. Define

$$\mathcal{O}_w^k = \{T \times K_k : T \in \mathcal{O}_w\} .$$

It follows from Lemma 14 that the upper bound of Theorem 12 is tight as the bound is attained by all graphs in $\mathcal{O}_w^k$. Further, because the graphs in $\mathcal{O}_w^k$ are minor-minimal, the bound of Theorem 12 is attained by all products of trees and cliques.

We have remarked that k-composition is a generalization of 3-star composition. Finally, we show that the above results on proper pathwidth of 3-star compositions of graphs can be strengthened to k-compositions.

COROLLARY 17. *Let* $G_0, \ldots, G_3$ *be* k*-connectable graphs, each of proper pathwidth at least* w*, and let* $G = G_0 \cup \cdots \cup G_3$*. Then,* $\mathbf{ppw}(G) \geq w + k$*.*

Proof. Let $n = |V(G)|$.

$$\begin{aligned}
\mathbf{ppw}(G) &= \mathbf{mvams}(G, 2n) && \text{(Corollary 8)} \\
&\geq \min_{0 \leq i \leq 3} \mathbf{mvams}(G_i, n) + k && \text{(Lemma 14)} \\
&= \min_{0 \leq i \leq 3} \mathbf{ppw}(G_i) + k && \text{(Corollary 8)} \\
&\geq w + k . \qquad \square
\end{aligned}$$

7. Conclusions and open problems. We have presented our results in the setting of mixed search (i.e., searching with placement, removal, and sliding of cops). For node search (searching with only placement and removal of cops), we can similarly define parameters $\mathbf{vans}(G)$ and $\mathbf{mvans}(G, r)$ for the general and monotone node search numbers for r visible, active robbers. Similarly, we can adapt all definitions of mixed-search parameters given in this paper to their node search counterparts.

The difference between mixed search and node search is not very great: node search can be reduced to mixed search, and the node search number is either equal to the corresponding mixed search number or one greater, depending on the graph.

We could, in principle, rewrite the present paper in terms of node search. Writing $\mathbf{pw}(G)$ and $\mathbf{tw}(G)$ for the well-known parameters of pathwidth and treewidth, it can be shown, using the results in [6, 11], that, for a graph of order n, $\mathbf{vians}(G,1) = \mathbf{tw}(G)+1$ and $\mathbf{vians}(G,n) = \mathbf{pw}(G)+1$. For completeness, we restate our core results for this setting:

$$\mathbf{mvans}(T,r) = \min\{\mathbf{pw}(T), \lfloor \log r \rfloor + 1\} + 1 \qquad \text{(for any tree } T\text{)},$$
$$\mathbf{mvans}(G,r) \leq \min\{\mathbf{pw}(G)+1, (\mathbf{tw}(G)+1)\cdot(\lfloor \log r \rfloor + 1)\} \qquad \text{(for any graph } G\text{)}.$$

Moreover, the framework of this paper can be applied to the other classical search variant, edge search. As this version can also be reduced to mixed searching (see, e.g., [3, 12]), we make no further comments in this direction.

The problem settled in this paper can be stated in the following way: given a graph G, what is the maximum number of visible, active robbers that can be captured by k cops? According to our results, this number is unbounded if $k \geq \mathbf{ppw}(G)$. In the case that $k < \mathbf{ppw}(G)$, the maximum number of robbers that can be caught in a tree is 2^{k-1}, and, for general graphs, it is at least $2^{k/\mathbf{ptw}(G)-1}$. This interpretation of our results may be useful for estimating how many sweeps of a graph a small number of cops needs to catch a large number of robbers.

We identify three main open problems on the study of graph searching for many robbers. The first is to find good lower bounds for $\mathbf{mvams}(G,r)$ in terms of G and r, for general graphs, corresponding to the bounds for trees found in this paper. We believe that this is a hard task as such a study appears to require the identification of obstructions for $\mathbf{mvams}(G,r)$ for all values of r.

Another open problem is to find graph decompositions corresponding to the game, tuning between (proper) tree decompositions (the case for one robber) and (proper) path decompositions (one robber per vertex). It is unclear what form such a family of decompositions would take.

Finally, it would be interesting to know whether there is any relation between our results and the search game defined by Fomin, Fraigniaud, and Nisse [8]. That game has only one robber but tunes between pathwidth and treewidth by limiting the number of rounds at which the cops may ask for the position of the robber. This provides an alternative way of tuning the interactivity of the game: it is fully interactive if the cops may ask for the robber's position at every move and fully predetermined if they may never ask for his position. Correspondingly, our game is fully interactive with a single robber and fully predetermined with a robber for each vertex of the graph.

REFERENCES

[1] B. Alspach, *Searching and sweeping graphs: A brief survey*, Matematiche (Catania), 59 (2004), pp. 5–37 (2006).

[2] D. Bienstock, *Graph searching, path-width, tree-width, and related problems (a survey)*, in Reliability of Computer and Communication Networks (New Brunswick, NJ, 1989), DIMACS Ser. Discrete Math. Theoret. Comput. Sci. 5, AMS, Providence, RI, 1991, pp. 33–49.

[3] D. Bienstock and P. Seymour, *Monotonicity in graph searching*, J. Algorithms, 12 (1991), pp. 239–245.

[4] B. Bollobás, *Extremal Graph Theory*, Dover, Mineola, NY, 2004. Reprint of the 1978 original.

[5] Y. Colin de Verdière, *Multiplicities of eigenvalues and tree-width of graphs*, J. Combin. Theory Ser. B, 74 (1998), pp. 121–146.

[6] N. D. Dendris, L. M. Kirousis, and D. M. Thilikos, *Fugitive-search games on graphs and related parameters*, Theoret. Comput. Sci., 172 (1997), pp. 233–254.

[7] J. A. Ellis, I. H. Sudborough, and J. S. Turner, *The vertex separation and search number of a graph*, Inform. and Comput., 113 (1994), pp. 50–79.

[8] F. V. Fomin, P. Fraigniaud, and N. Nisse, *Nondeterministic graph searching: From pathwidth to treewidth*, in Mathematical Foundations of Computer Science 2005, Lecture Notes in Comput. Sci. 3618, Springer-Verlag, Berlin, 2005, pp. 364–375.

[9] F. V. Fomin and D. M. Thilikos, *Multiple Edges Matter When Searching a Graph: The Exact Figure*, manuscript.

[10] P. Hunter and S. Kreutzer, *Digraph measures: Kelly decompositions, games, and orderings*, in Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms (New Orleans, 2007), ACM, New York, SIAM, Philadelphia, 2007, pp. 637–644.

[11] P. D. Seymour and R. Thomas, *Graph searching and a min-max theorem for tree-width*, J. Combin. Theory Ser. B, 58 (1993), pp. 22–33.

[12] Y. C. Stamatiou and D. M. Thilikos, *Monotonicity and inert fugitive search games*, Electronic Notes in Discrete Mathematics, 3 (1999).

[13] A. Takahashi, S. Ueno, and Y. Kajitani, *Minimal acyclic forbidden minors for the family of graphs with bounded path-width*, Discrete Math., 127 (1994), pp. 293–304.

[14] A. Takahashi, S. Ueno, and Y. Kajitani, *Mixed searching and proper-path-width*, Theoret. Comput. Sci., 137 (1995), pp. 253–268.

[15] D. M. Thilikos, *Algorithms and obstructions for linear-width and related search parameters*, Discrete Appl. Math., 105 (2000), pp. 239–271.