

Locating a robber on a graph via distance queries

Lawrence H. Erickson*, James Carraher†, Ilkyoo Choi‡, Michelle Delcourt§,
Douglas B. West¶

October 11, 2011

Abstract

A cop wants to locate a robber hiding among the vertices of a graph. A round of the game consists of the robber moving to a neighbor of its current vertex (or not moving) and then the cop scanning some vertex to obtain the distance from it to the robber. If the cop can at some point determine where the robber is, then the cop wins. If the robber can avoid this forever, then the robber wins. We prove that the robber wins on graphs with girth at most 5. We also prove a conjecture of Seager by showing that the cop wins on a subdivision of an n -vertex graph G when each edge is subdivided into a path of length at least $\min\{3^{\mu(G)}, n\}$, where $\mu(G)$ denotes the minimum size of a vertex subset W such that the vertices of G have distinct vectors of distances to the vertices of W . For grids and complete bipartite graphs, the threshold is lower.

Keywords: graph searching; cops and robbers; metric dimension; resolving set; subdivision

1 Introduction

We study a pursuit-evasion game on a graph H . The evader (the *robber*) moves among the vertices of the graph, moving distance 0 or 1 in each round. After a robber move, the pursuer (the *cop*) scans a vertex in the graph and obtains the distance from it to the vertex where

*Department of Computer Science, University of Illinois at Urbana-Champaign, lericks4@uiuc.edu. Partially supported by NSF grant 0904501 (IIS Robotics), NSF grant 1035345 (CNS Cyberphysical Systems), DARPA STOMP grant HR0011-05-1-0008, and MURI/ONR grant N00014-09-1-1052.

†Department of Mathematics, University of Nebraska-Lincoln, s-jcarrah1@math.unl.edu. Partially supported by NSF grant DMS 0838434 EMSW21MCTP: Research Experience for Graduate Students and NSF grant DMS 0914815.

‡Department of Mathematics, University of Illinois at Urbana-Champaign, ichoi4@illinois.edu.

§Department of Mathematics, University of Illinois at Urbana-Champaign, delcour2@illinois.edu. Partially supported by NSF grant DMS 0838434 EMSW21MCTP: Research Experience for Graduate Students and the NSF Graduate Research Fellowship DGE 1144245.

¶Department of Mathematics, University of Illinois at Urbana-Champaign, west@math.uiuc.edu. Partially supported by NSA grant H98230-10-1-0363.

the robber is. If the cop can then determine where the robber is, the cop wins. If the robber can avoid this forever, then the robber wins. The cop is not required to scan the vertex containing the robber in order to win. A robber move and subsequent cop scan together form a *round* of the game.

Our model is similar to that of Seager [11],[12]. Seager imposes the additional restriction that the robber cannot move to the vertex that was scanned in the previous round. We call this the *no-backtrack condition*; it makes the cop stronger. Hence our conditions for the cop to win are also sufficient in Seager’s model. Our results that do not apply to Seager’s model are those in Section 5 showing that the robber wins on graphs with small cycles.

Cops-and-robbers games on graphs were introduced by Parsons in [10] and have been studied extensively under various names (searching, sweeping, clearing, hunter/rabbit, and others). Parsons’ original problem involves an omniscient evader and some number of pursuers who receive no information about the evader’s location unless they occupy the same vertex as the evader. A variant in which the cop and robber always know each other’s locations and the cop must land on the robber to win is studied in [9]. Another variant allowing the robber to move any distance along a path containing no cop is related to the tree-width of the graph in [13]. A randomized algorithm for cops and robbers with limited visibility is described in [5]. Surveys detailing other variants include [1] and [3].

In contrast to most of these models, in our problem the cop has no location in the graph. The cop may scan any vertex. Winning only means that the cop has restricted the robber’s possible locations to a single vertex; the “troops” can then go apprehend the robber. Practical applications of models involving distance information may arise in the setting of sensors that are embedded at various locations and can be queried (see [7]).

Seager [11],[12] proved that under the no-backtrack condition, the cop wins on paths, cycles other than the 5-cycle, and trees. She also proved structural theorems: the robber wins on any graph containing K_4 as a subgraph or $K_{3,3}$ as an induced subgraph. She conjectured that for any graph G , the cop wins on some subdivision of G .

We prove this conjecture. The *m-subdivision* of a graph G is the graph obtained from G by replacing each edge of G with a path of length m through new vertices of degree 2; the notation for the resulting graph is $G^{1/m}$. We prove that if G has n vertices, then the cop wins on $G^{1/m}$ when $m \geq \min\{3^{\mu(G)}, n\}$, where $\mu(G)$ is the metric dimension of G . To define metric dimension, note that for $W = \{w_1, \dots, w_k\} \subseteq V(G)$ and $v \in V(G)$, the distances from v to the members of W form a k -tuple $(d(v, w_1), \dots, d(v, w_k))$. If no two vertices of $V(G)$ yield the same k -tuple, then W is a *resolving set* for G . The *metric dimension* $\mu(G)$ is the minimum size of a resolving set for G .

The metric dimension of a graph was introduced independently by Slater [14] and by Harary and Melter [4]. Computing it is NP-complete [6]. The only graphs with metric dimension 1 are paths, and the only n -vertex graphs with metric dimension $n-1$ are complete graphs. The n -vertex graphs with metric dimension $n-2$ have also been determined [2].

Using metric dimension to resolve Seager’s conjecture is natural. If the robber cannot move, then scanning a resolving set of G finds the robber. Subdividing edges into sufficiently long paths “slows down” the robber enough that scanning a resolving set of G still provides useful information about the robber’s location in $G^{1/m}$. (To motivate the notation $G^{1/m}$, note that m -subdivision pushes vertices of G farther apart, whereas the distance power G^m adds edges to reduce distances between vertices of G .)

Section 2 shows that the cop wins on $G^{1/m}$ when $m \geq \min\{3^{\mu(G)}, n\}$. Seager’s conjecture follows because her cop is stronger than ours. We prove separately that $m \geq 3^{\mu(G)}$ and $m \geq n$ each suffice. Sections 3 and 4 reduce this threshold on m for a cop win when G is a complete bipartite graph or a grid. Section 5 shows that the robber wins on any graph having a cycle of length at most 5. Section 6 discusses open problems and conjectures.

2 Proof of Seager’s Conjecture

A *thread* in a graph G is a path in G whose internal vertices have degree 2 in G ; thus m -subdivision replaces each edge by a thread of length m . A robber known to be in a thread can be located in one move by scanning one of its endpoints. Also any subdivision of a path is a path. Hence we may assume that the given graph G is not a path. We may also restrict our attention to connected graphs, since it is easy to determine which component contains the robber. When G is not a path, $\mu(G) \geq 2$ (see [2]).

Definition 2.1. Given a connected graph G that is not a path, let $m \geq 3^{\mu(G)}$, and let $H = G^{1/m}$. Let $\hat{V}(H) = V(H) \cap V(G)$; the set $\hat{V}(H)$ consists of the vertices that were present prior to subdividing edges. For $v \in \hat{V}(H)$, let the *span* of v , written $\text{Span}(v)$, be $\{z \in V(H) : d_H(v, z) \leq m - 1\}$, where $d_H(x, y)$ denotes the distance in H between vertices x and y . When a vertex $q \in V(H)$ is scanned by the cop, let $\sigma(q)$ be the result of the scan (the distance from q to the robber at that time). For vertices $v_i, v_j \in \hat{V}(H)$ such that $v_i v_j \in E(G)$, let $P(v_i, v_j)$ denote the path of length m in H having v_i and v_j as endpoints.

Let $W(G)$ denote a fixed smallest resolving set of G , and name its vertices $w_1, \dots, w_{\mu(G)}$. For $H = G^{1/m}$, let $\hat{W}(H) = W(G) \cap \hat{V}(H)$. For $v \in \hat{V}(H)$, the W -distance vector of v is the $\mu(G)$ -tuple whose i th entry is $d_G(v, w_i)$.

Lemma 2.2. *With $H = G^{1/m}$, any m consecutive positions of the robber in H lie in the span of some vertex $v \in \hat{V}(H)$.*

Proof. Let S be the set of positions occupied by the robber during some m consecutive rounds. If $v \in S$ for some $v \in \hat{V}(H)$, then $S \subseteq \text{Span}(v)$, since S contains at most m distinct vertices, none of which can have distance more than $m - 1$ from v .

If S does not intersect $\hat{V}(H)$, then S is contained among the internal vertices of some thread joining vertices $v_i, v_j \in \hat{V}(H)$. In this case, S lies in the span of both v_i and v_j . \square

When the vertices $w_1, \dots, w_{\mu(G)}$ corresponding to the resolving set $W(G)$ are scanned consecutively, we obtain a $\mu(G)$ -tuple of distances from vertices of $\hat{W}(H)$ to the robber at successive times; call it D . Since $\hat{W}(H)$ is a resolving set of G , the W -distance vectors of vertices in G are distinct. If the robber sits at a vertex $z \in \hat{V}(H)$ during this time, then $D = (md_G(z, w_1), \dots, md_G(z, w_{\mu(G)}))$. In fact, the robber can move, but by Lemma 2.2 the robber remains within the span of some vertex v in $\hat{V}(H)$.

Lemma 2.3. *Let D be the $\mu(G)$ -vector of distances to the robber returned by scanning the vertices of $\hat{W}(H)$ in succession. If the robber stays in $\text{Span}(v)$ for some $v \in \hat{V}(H)$ during this scan, then*

$$D = (md_G(v, w_1) + \rho_1, \dots, md_G(v, w_{\mu(G)}) + \rho_{\mu(G)})$$

for constants $\rho_1, \dots, \rho_{\mu(G)}$ satisfying $-(m-1) \leq \rho_i \leq m-1$.

Proof. When $z \in \text{Span}(v)$, we have $md_G(v, w_i) - m + 1 \leq d_H(z, w_i) \leq md_G(v, w_i) + m - 1$, by the triangle inequality. \square

A *candidate* is a vertex $v \in \hat{V}(H)$ satisfying the *conclusion* of Lemma 2.3; that is, the vector of distances generated during the scan of $\hat{W}(H)$ can be modified by at most $m-1$ in each coordinate to become the W -distance vector of v . By Lemma 2.3, such a vertex is a “candidate” to be a vertex whose span contains the robber during the scan of $\hat{W}(H)$. Lemmas 2.2 and 2.3 together imply that there is at least one candidate; our next lemma leads to an upper bound on the number of candidates.

Lemma 2.4. *If $|d_G(w, v_1) - d_G(w, v_2)| \geq 2$ for some vertex $w \in W(G)$ and vertices $v_1, v_2 \in \hat{V}(H)$, then v_1 and v_2 are not both candidates.*

Proof. Index v_1 and v_2 so that $d_G(w, v_1) < d_G(w, v_2)$. Since $d_G(w, v_2) - d_G(w, v_1) \geq 2$ and edges of G become threads of length m in H , it follows that $d_H(w, v_2) \geq d_H(w, v_1) + 2m$. Let x be the position of the robber when w is scanned. If both v_1 and v_2 are candidates, then $d_H(w, v_2) - (m-1) \leq d_H(w, x) \leq d_H(w, v_1) + m - 1$, a contradiction. \square

Corollary 2.5. *There are at most $2^{\mu(G)}$ candidates.*

Proof. Letting D_i be the value in the i th position of D , we have $D_i = md_G(w_i, v) + \rho_i$ when v is a candidate. Given that $-(m-1) \leq \rho_i \leq m-1$, there is only one integer a such that $D_i = ma + b = m(a+1) - (m-b)$ for a nonnegative integer b . Therefore, for each i there are only two possible values for $d_G(w_i, v)$.

Since W is a resolving set in G , the $2^{\mu(G)}$ resulting possible W -distance vectors for candidates determine at most one vertex each. \square

Finally, we bound the maximum degree $\Delta(G)$ in terms of the size of $W(G)$.

Lemma 2.6. *For a graph G , $\Delta(G) < 3^{\mu(G)}$.*

Proof. The W -distance vectors of neighboring vertices of G differ by at most 1 in each position. Hence there are fewer than $3^{\mu(G)}$ distance vectors of neighbors of a vertex v . Since W is a resolving set, each such vector arises for at most one vertex of G . Hence v has fewer than $3^{\mu(G)}$ neighbors. \square

We present an algorithm to locate the robber in $G^{1/m}$ when $m \geq 3^{\mu(G)}$. It has several phases. First it finds candidates and then a particular vertex $z \in \hat{V}(H)$ whose span contains the robber. It then scans the neighbors of z in G until it locates the robber or finds a new vertex $z' \in \hat{V}(H)$ that is closer to the robber than z was when it was chosen. This process is repeated fewer than m times and can only end by locating the robber or by scanning the vertex where the robber is.

Theorem 2.7. *If $H = G^{1/m}$ with $m \geq 3^{\mu(G)}$, then the cop can locate the robber in H within $m(3^{\mu(G)} - 1)$ rounds.*

Proof. Playing as the cop, we describe the strategy and simultaneously argue that it works.

The first phase is ScanW. We scan all of $\hat{W}(H)$ to produce a set C of candidates, at least one of which (by Lemma 2.2) contains the robber in its span throughout this phase. Lemma 2.4 implies that $|C| \leq 2^{\mu(G)}$.

Next, in the PickOne phase, vertices of C are scanned until a vertex $z \in C$ is scanned such that $\sigma(z) < m$. Since $m \geq 3^{\mu(G)} > \mu(G) + 2^{\mu(G)}$ when $\mu(G) \geq 2$, Lemma 2.2 implies that the robber remains in the span of one vertex of $\hat{V}(H)$ from the beginning of the ScanW phase until all of C (if needed) is scanned. By Lemma 2.3, each such vertex is a candidate. Therefore, the PickOne phase does scan a candidate z that returns $\sigma(z) \leq m - 1$. Let ρ be that distance at that time; at this point we leave the PickOne phase.

Having specified z and ρ , we enter the GetCloser phase. Let $T = \{v \in \hat{V}(H): d_H(v, z) = m\}$. Note that T is the neighborhood in G of z , so $|T| < 3^{\mu(G)}$ by Lemma 2.6. We scan vertices of T until we locate the robber or scan a vertex $z' \in T$ such that $\sigma(z') < \rho$. If $\sigma(z') = 0$, then we have found the robber. Otherwise, z' becomes z , the shorter distance becomes ρ , and we restart the GetCloser phase with this better candidate. Each trip through the GetCloser phase scans fewer than $3^{\mu(G)}$ vertices; we must show that it locates the robber or produces the claimed vertex z' .

Let S denote the set of positions held by the robber while T is being scanned. Suppose first that S contains a vertex v of T . Since $|T| < m$, we have $S \subseteq \text{Span}(v)$. Also, the robber cannot reach v until at least $m - \rho$ rounds into this GetCloser phase, since at the start of the phase the robber has distance ρ from z . Again since $|T| < m$, fewer than ρ vertices are scanned in this phase after the robber reaches v . If v is scanned sometime after $m - \rho - 1$ rounds of this phase, then $\sigma(v) < \rho$ when v is scanned. Hence in this case v is the desired vertex z' .

If v is scanned after at most $m - \rho - 1$ rounds in this phase, then the robber is still in $\text{Span}(z)$, since it takes at least $m - \rho$ steps to leave $\text{Span}(z)$. Since $\text{Span}(z) \cap \text{Span}(v) \subseteq P(z, v)$, the value of $\sigma(v)$ uniquely determines the robber's position when v is scanned.

Hence we may assume that $S \cap T = \emptyset$. This keeps the robber within $\text{Span}(z)$ during the scan of T . Scanning $v \in T$ yields $\sigma(v)$ divisible by m if and only if the robber is then in $T \cup \{z\}$. If $z \notin S$, then no scan in this phase gives distance divisible by m , and we know that the robber remains in the original thread, $P(z, v)$ for some $v \in T$. When we scan v , the robber is known to be on $P(z, v)$, and the value of $\sigma(v)$ locates the robber.

The final case is $S \cap T = \emptyset$ and $z \in S$. Let $M(z) = \{x \in V(H) : \lfloor m/2 \rfloor \leq d_H(z, x) \leq \lceil m/2 \rceil\}$; we call $M(z)$ the *midway set* of z . The robber is in $M(z)$ during this phase when the scan distance is congruent to $\lfloor m/2 \rfloor$ or $\lceil m/2 \rceil$ modulo m (if the robber has not left $\text{Span}(z)$). If the scan yields a multiple of m sometime before $M(z)$ is entered, then we know whether the robber is at z or in T then, since we know whether ρ is large or small. If at z , then we have located the robber; if in T , then $S \cap T \neq \emptyset$ and an earlier case applies.

Until the robber visits $M(z)$, the scan of a vertex $v \in T$ determines whether the robber moved toward z , away from z , or did not move, since ρ determines that the robber is between $M(z)$ and z , and the congruence class modulo m of the distance to the robber determines whether the shortest path to the robber from v goes through z .

If the robber visits $M(z)$ and later z , then when the scan returns a multiple of m we may not know whether the robber is at z or in T (we may not learn that $S \cap T \neq \emptyset$ until a later scan). Figure 1 shows examples; vertices in grey on the left are in $M(z)$ (each thread at z has two such vertices when m is odd). If the vertex indicated by an arrow is now scanned and the distance is $m + \lfloor (m - 1)/2 \rfloor$, then it cannot be determined whether the robber moved toward or away from z .

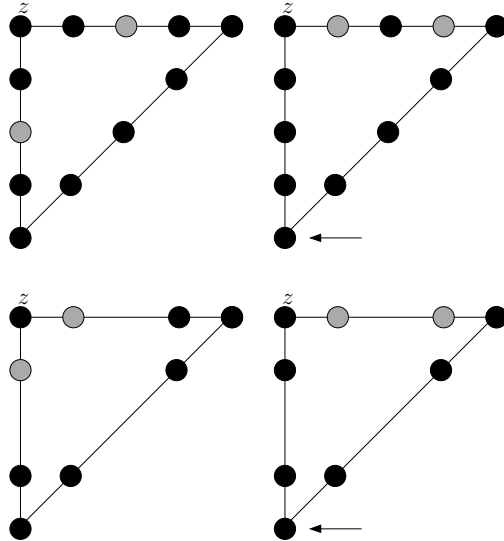


Figure 1: Scanning a vertex after the robber was in the midway set $M(z)$.

Suppose that m is even. After visiting $M(z)$, which took at least $|\rho - m/2|$ rounds of this phase, at least $m/2$ rounds are spent to reach z . Since $|T| < m$, not enough rounds remain in the phase for the robber to go farther than $\rho - 1$ steps away from z . Hence if we do not learn that $S \cap T \neq \emptyset$ by the end of the phase, we must be in this case: $S \cap T = \emptyset$, and z was visited. We then repeat the GetCloser phase using the same vertex z and a distance, computed from the result of scanning the last vertex of T , that is smaller than ρ .

When m is odd, the robber must spend two consecutive rounds in $M(z)$ to create ambiguity. Reaching $M(z)$ takes at least $|\rho - m/2| - 1/2$ rounds. If the scan distance is a multiple of m after the fewest possible additional rounds, then the robber is at z if $\rho < m/2$ and in T if $\rho > m/2$. Creating ambiguity thus requires an extra round in $M(z)$. Now, as in the even case, the robber cannot go farther than $\rho - 1$ steps away from z before all of T has been scanned. The phase then concludes in the same way as for even m .

We have shown that in all cases the robber is located or a new GetCloser phase is initiated with a smaller value of ρ . Since ρ is at most $m - 1$ to start the first GetCloser phase, this phase repeats at most $m - 1$ times. Each such phase takes fewer than $3^{\mu(G)}$ rounds. The ScanW and PickOne phases together also take fewer than $3^{\mu(G)}$ rounds. Therefore, the entire algorithm takes at most $m(3^{\mu(G)} - 1)$ rounds to find the robber. \square

Some graphs have metric dimension nearly as large as the vertex set. For example, the complete graph K_n has metric dimension $n - 1$, and there are three families of n -vertex graphs with metric dimension $n - 2$ [2]. The argument above used $\mu(G)$ to obtain bounds on the number of candidates and on the maximum degree of G . When G has n vertices, trivially the number of candidates is at most n and $\Delta(G) \leq n - 1$. Using these bounds, we obtain a similar algorithm when m is at least $|V(G)|$.

Theorem 2.8. *If G is an n -vertex graph, and $H = G^{1/m}$ with $m \geq n$, then on H the cop can locate the robber within $m(n - 1) + 1$ rounds.*

Proof. We use a variant of the previous algorithm. The resolving set of G does not need to be scanned, as we let every vertex in $\hat{V}(H)$ be a candidate. In essence, we jump directly to the PickOne phase and plan to scan all of $\hat{V}(H)$ during the first n rounds.

If the robber visits a vertex v in $\hat{V}(H)$ during this phase, then the robber remains in $\text{Span}(v)$ during the first n rounds, since $n \leq m$. The scan of v returns $\sigma(v) \leq m - 1$.

If the robber does not enter $\hat{V}(H)$ during the first n moves, then the robber stays in a thread $P(v, v')$ for some v and v' , with v' later in the scan list than v . Again $\sigma(v) \leq m - 1$.

In either case, during the first n rounds we scan a vertex v with $\sigma(v) \leq m - 1$. At that time, we start the first GetCloser phase with v being z and $\sigma(v)$ being ρ . Since $|T| < n$, each GetCloser phase takes fewer than n rounds. The initial PickOne phase may take n rounds, so we have the upper bound $m(n - 1) + 1$. \square

Theorems 2.7 and 2.8 provide the following corollary.

Corollary 2.9. *If G is an n -vertex graph, and $H = G^{1/m}$, then the cop can locate the robber on H if $m \geq \min\{3^{\mu(G)}, n\}$.*

3 Subdivisions of Complete Bipartite Graphs

When G is a complete bipartite graph, the threshold on m for the cop to win in $G^{1/m}$ is smaller than Corollary 2.9 would indicate. In this section $\text{Span}(v)$ is defined slightly differently: let $\text{Span}(v) = \{p \in V(H) : d(p, v) \leq m\}$ when $H = G^{1/m}$ and $v \in \hat{V}(H)$.

Theorem 3.1. *If $H = K_{a,b}^{1/m}$ with $m \geq a \geq b$ and $m > b$, then the cop can locate a robber on H within $b + m(b + a)$ rounds.*

Proof. Let $G = K_{a,b}$. Let S and T be the partite sets of G , with $|S| = a$ and $|T| = b$. We play a cop strategy. In the first phase, called BiPickOne, we scan vertices of T until finding a vertex $t \in T$ with $\sigma(t) \leq m$. If the robber moves into S , then BiPickOne ends on the scan for that round. If the robber does not move into S , then the robber remains in the span of one vertex $t \in T$ during the phase, and $\sigma(t) \leq m$ when t is scanned. Let ρ be the distance from the scanned vertex to the robber when BiPickOne ends.

We next enter the ScanS phase, scanning vertices of S until the robber is located or the distance returned by the scan is at most ρ . In the latter case, we name this vertex z , let $\sigma(z)$ be the new value of ρ , and switch to ScanT.

Because $|S| = a \leq m$, the ScanS phase lasts at most m rounds. If $\rho = m$ when ScanS starts, then the robber was at some vertex $s \in S$ at the end of ScanT. When s is scanned in ScanS, we obtain $\sigma(s) \leq m$, and s will be chosen as z . If $\rho < m$, then the only vertex of T that the robber has time to reach during ScanS is t ; if the robber returns to t , then the distance at that scan is m and the robber is located at t .

We may therefore assume that $\rho < m$ and that the robber does not return to T . In this case, the robber remains in the span of some vertex $s \in S$ during the entire ScanS phase. Until the scan distance is a multiple of m , the robber has not entered s and is still in the thread joining s and t . If s is scanned before the scan distance has been a multiple of m , then scanning s locates the robber.

If s is scanned after the scan distance has been a multiple of m , then it took at least $m - \rho$ moves for the robber to reach s , so $\sigma(s) \leq \rho$ when s is scanned. We let s be z , use $\sigma(s)$ as the new value of ρ , and enter ScanT.

The ScanT phase is almost identical to the ScanS phase. Vertices of T are scanned until the robber is located or scanning some $t \in T$ yields $\sigma(t) < \rho$. Because $|T| = b < m$, this phase lasts at most $m - 1$ rounds. The arguments are the same as above with S and T switched, except that $|T| < m$ results in a guarantee that $\sigma(t) < \rho$ rather than $\sigma(t) \leq \rho$ when we fall into the cases that kick us into a scan of the other partite set.

The BiPickOne phase takes at most b rounds and produces a value of ρ that is at most m . Each call to ScanS and then ScanT that does not locate the robber drops the value of ρ by at least 1 and scans each vertex at most once. Hence there are at most $b + m(a + b)$ rounds until the cop locates the robber. \square

4 Subdivisions of Grids

The *grid* $G_{k,l}$ is the graph with vertex set $\{v_{i,j}: 1 \leq i \leq k, 1 \leq j \leq l\}$ defined by making $v_{i,j}$ and $v_{i',j'}$ adjacent if and only if $|i - i'| + |j - j'| = 1$ (see Figure 2). When $k, l \geq 2$, the graph $G_{k,l}$ contains 4-cycles. We will show (in Section 5) that the robber wins on such graphs. However, subdividing each edge of $G_{k,l}$ once produces a graph on which the cop wins. A very special case of various results in [6] and [8] is that grids have metric dimension 2 ($\{v_{1,1}, v_{k,1}\}$ is a resolving set); Theorem 2.7 thus implies that the cop wins on $G_{k,l}^{1/m}$ when $m \geq 9$. Here we lower the threshold to 2.

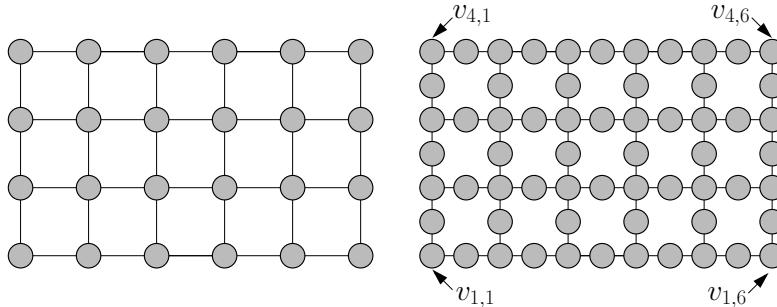


Figure 2: The graph $G_{4,6}$ and the graph $G_{4,6}^{1/2}$.

Theorem 4.1. *If $H = G_{k,l}^{1/m}$ with $m \geq 2$, then the cop can locate the robber on H within three rounds.*

Proof. First scan $v_{1,1}$. If $\sigma(v_{1,1})$ is a multiple of m , then the robber is in $\hat{V}(H)$ at the time of the scan, at a vertex in $\{v_{i,j}: i + j - 2 = \sigma(v_{1,1})/m\}$; call this set S (Figure 3 top left). After moving in round 2, the robber is at a vertex in the set S' consisting of all vertices within distance 1 of a vertex in S . The possible locations group into disjoint sets of five vertices around vertices of S , except for smaller sets around the elements of S with degree less than 4 (Figure 3 top right).

The cop now scans the (unique) vertex of S' closest to $v_{k,1}$. The subsets of S' with a fixed distance from $v_{k,1}$ have size 1 consisting of a vertex in $\hat{V}(H)$ or size 2 consisting of two neighbors of such a vertex. The robber sits in one such set, already located unless the set has size 2 (Figure 3 bottom left).

In that case, the robber moves in round 3. The robber can remain still, move back to its original location in $\hat{V}(H)$, or move one step farther away from it. The resulting five possible

locations form a path in H , and distances along the path are true distances in H . Now the cop scans an endpoint of this path to locate the robber (Figure 3 bottom right).

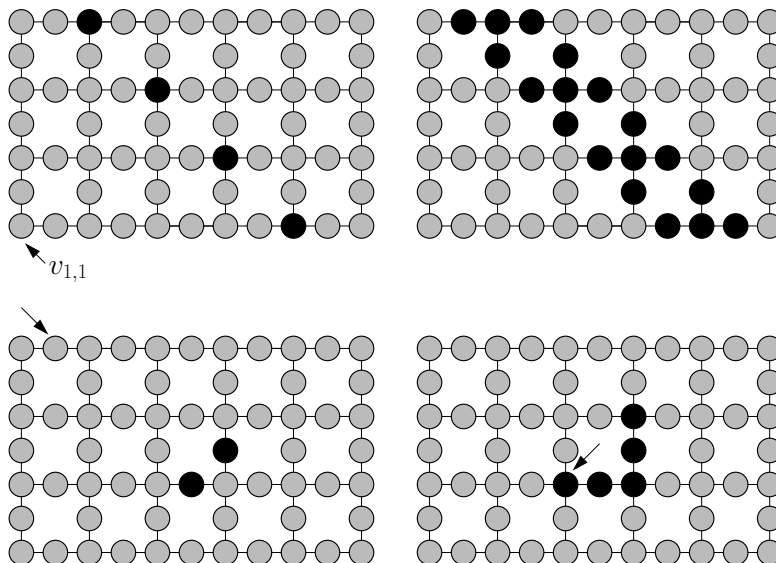


Figure 3: The game on the grid when $\sigma(v_{1,1})$ is a multiple of m .

When $\sigma(v_{1,1})$ is not a multiple of m , let $a = \lfloor \sigma(v_{1,1})/m \rfloor$ and $b = \sigma(v_{1,1}) - ma$. The robber is initially at a vertex of degree 2. The possible vertices again form a “diagonal” in the drawing of H , with fixed distance b from the vertices that would be the candidates if $\sigma(v_{1,1})$ had been equal to ma (Figure 4 left).

In round 2, the robber moves at most 1 step along the thread containing the original location. The key observation is that all possible locations of the robber now lie along a path; when $m = 2$, they induce a path (Figure 4 right). Again distances along the path are distances in H , so the cop locates the robber by scanning an endpoint of this path. \square

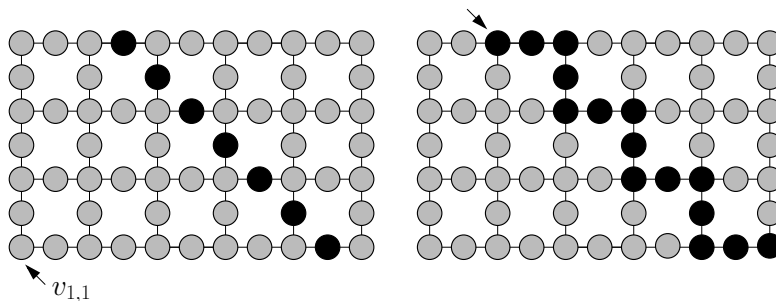


Figure 4: The game on the grid when $\sigma(v_{1,1})$ is not a multiple of m .

5 Graphs with Short Cycles

The cop wins on the n -vertex cycle C_n if $n \geq 7$, even without the no-backtrack condition [11]. However, the robber wins on graphs having small cycles. When π and τ are partitions of a set S , we say that π is a *refinement* of τ if every block of π is contained in one block of τ .

Here we must clarify the meaning of “robber wins”. There is no graph on which the robber can guarantee not being located, since the cop may get lucky and sit on the robber. Hence the meaning of “robber wins” is that the cop cannot guarantee winning.

Theorem 5.1. *If a graph G has a cycle of length at most 5, then the robber wins on G .*

Proof. Let C be a shortest cycle in G . The robber magnanimously agrees to stay within $V(C)$. Even with this restriction on the robber’s movement, the cop will not be able to locate the robber.

A scan of any vertex v partitions $V(C)$ into sets by their distance from v in G ; the robber is then known to lie in one such set. If the partition given by scanning u is a refinement of the partition given by v , then the cop gains more information by scanning u . When C has odd length, the most refined partitions of $V(C)$ that can be obtained by scans put one vertex u by itself and pair vertices of C having equal distance along C from u (see Figure 5). Such a partition arises by scanning u , for example.

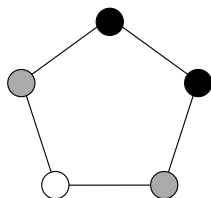


Figure 5: The sole non-refinable partition of a 5-cycle, up to automorphism.

When C has length 4, scanning a vertex u of C yields an unrefinable partition in which the neighbors of u are in the same set, but u and the opposite vertex are alone. Another unrefinable partition may arise by scanning a vertex not on C ; the blocks are two pairs of consecutive vertices on C (see Figure 6). To see that these are the unrefinable partitions, recall that the distances from z to adjacent vertices differ by at most 1. Hence if three consecutive vertices lie in distinct parts, partition 1 in Figure 6) is forced. Otherwise, if blocks of the partition have size at most 2 and there is a consecutive pair, the distances from z to the remaining vertices must differ from this by 1, and since they are adjacent the change must be in the same direction; this yields partition 2. The analysis for odd cycles is similar and simpler.

We make the cop strongest by letting the cop impose any of these unrefinable partitions on C with any scan. The first scan restricts the robber to a set of size 2; otherwise the robber is located. The robber can then move. If the set of size 2 dominates $V(C)$, then the

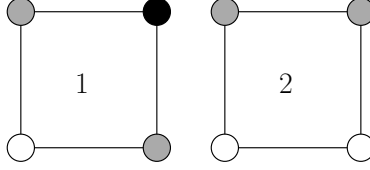


Figure 6: The non-refinable partitions of a 4-cycle, up to automorphism.

robber may now be anywhere in $V(C)$, and the cop has not obtained any information. The game is the same when the cop is ready to scan again as it was in round 1.

All the sets of size 2 in the unrefinable distance partitions of C have this domination property except the block of two adjacent vertices in a partition when C is a 5-cycle. In this case, after the next robber move the robber can be at any of four consecutive vertices on C . Every partition that the cop can next impose has two of those four vertices in the same block. Since the cop is unlucky, the robber may be at one of those two vertices. Whether they are adjacent or nonadjacent, we have again reached an earlier situation, so the cop cannot guarantee making any progress in locating the robber. \square

Seager [11] showed that the cop wins on a 6-cycle under the no-backtrack condition, but the cop does not win on a 6-cycle when that condition is dropped. In the model without that condition, we have not been able to determine whether the robber wins on every graph whose shortest cycles have length 6.

6 Conclusion and Open Problems

One cannot generally find a vertex in $G^{1/m}$ whose span contains the robber just by scanning the vertices of a resolving set of G . To show this, we construct a family of graphs. To form J_k , begin with a complete graph having 2^k vertices; their names are the binary k -tuples. Add an independent set S of size k corresponding to the coordinates. Make the i th vertex of S adjacent to the original vertices whose names have a 1 in coordinate i . This graph is J_k . Note that J_k has $k + 2^k$ vertices and that S is a resolving set, so $\mu(J_k) \leq k$.

Let $H = J_k^{1/m}$. If m is even, then scanning the set S may leave a potential robber location in the span of each vertex of $\hat{V}(H)$ that was a member of the K_{2^k} subgraph in J_k . Figure 7 shows an example for $k = 2$ and $m = 4$. The grey vertices form a resolving set of J_2 , and the white vertices show the possible robber locations when scanning the grey vertices yields distances 6 and 6.

Despite the existence of graphs like J_k , we believe there exists some function $f(n)$ such that (1) $f(n) = O(n)$ and (2) for any n -vertex graph G , the cop wins on $G^{1/f(n)}$. We further conjecture that the cop wins on $K_n^{1/m}$ if and only if $m \geq n$. If true, this conjecture would demonstrate sharpness on Theorem 2.8.

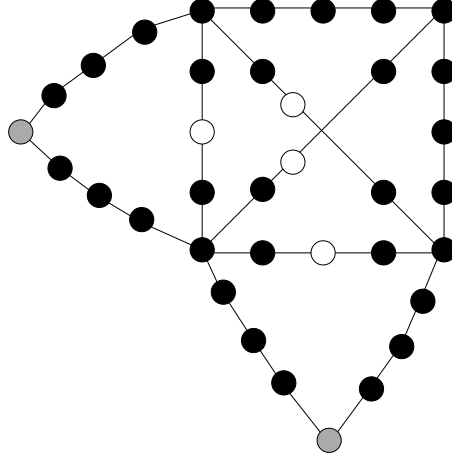


Figure 7: The graph $J_2^{1/4}$.

Our results used the fact that the edges of the original graph were subdivided into paths of equal length. We conjecture that if the cop wins on a graph G , then subdividing one edge of G produces another graph on which the cop wins. An affirmative answer could significantly simplify the understanding of the graphs on which the cop wins.

References

- [1] B. Alspach, “Searching and sweeping graphs: a brief survey,” *Le Matematiche*, vol. 59, no. 1–2, 2004.
- [2] G. Chartrand, L. Eroh, M. A. Johnson, and O. R. Oellermann, “Resolvability in graphs and the metric dimension of a graph,” *Discrete Applied Mathematics*, vol. 105, no. 1–3, pp. 99–113, 2000.
- [3] F. V. Fomin and D. M. Thilikos, “An annotated bibliography on guaranteed graph searching,” *Theoretical Computer Science*, vol. 399, no. 3, pp. 236 – 245, 2008.
- [4] F. Harary and R. A. Melter, “On the metric dimension of a graph,” *Ars Combinatorica*, vol. 2, pp. 191–195, 1976.
- [5] V. Isler, S. Kannan, and S. Khanna, “Randomized pursuit-evasion with local visibility,” *SIAM Journal on Discrete Mathematics*, vol. 1, no. 20, pp. 26–41, 2006.
- [6] S. Khuller, B. Raghavachari, and A. Rosenfeld, “Landmarks in graphs,” *Discrete Applied Mathematics*, vol. 70, no. 3, pp. 217–229, 1996.
- [7] S. M. LaValle, “Sensing and filtering: A tutorial based on preimages and information spaces,” *Foundations and Trends in Robotics*, 2011, to appear.

- [8] R. A. Melter and I. Tomescu, “Metric bases in digital geometry” *Computer Vision, Graphics, and Image Processing*, vol. 25, no. 1, pp. 113–121, 1984.
- [9] R. Nowakoski and P. Winkler, “Vertex-to-vertex pursuit in a graph,” *Discrete Math*, vol. 43, pp. 235–239, 1983.
- [10] T. Parsons, “Pursuit-evasion in a graph,” in *Theory and Applications of Graphs*, ser. Lecture Notes in Mathematics, Y. Alavi and D. Lick, Eds. Springer Berlin / Heidelberg, 1978, vol. 642, pp. 426–441.
- [11] S. Seager, “Locating a robber on a graph,” 2011, preprint.
- [12] S. Seager, “Locating a robber on a graph,” Presentation at 3rd CanaDAM Conference, Victoria, BC, June 1, 2011.
- [13] P. D. Seymour and R. Thomas, “Graph searching and a min-max theorem for tree-width,” *Journal of Combinatorial Theory, Series B*, vol. 58, no. 1, pp. 22–33, 1993.
- [14] P. J. Slater, “Leaves of trees,” in *Proceedings of the Sixth Southeastern Conference on Combinatorics, Graph Theory, and Computing*, 1975, pp. 549–559.