Theoretical Computer Science 463 (2012) 54–61

Contents lists available at SciVerse ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Locating a robber on a graph via distance queries

James Carraher[a], Ilkyoo Choi[b], Michelle Delcourt[b], Lawrence H. Erickson[c], Douglas B. West[b,*]

[a] Department of Mathematics, University of Nebraska-Lincoln, United States

[b] Department of Mathematics, University of Illinois at Urbana-Champaign, United States

[c] Department of Computer Science, University of Illinois at Urbana-Champaign, United States

ARTICLE INFO

Keywords:
Graph searching
Cops and robbers
Metric dimension
Resolving set
Subdivision

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 that vertex to the robber. If the cop can at some point determine where the robber is, then the cop wins; otherwise, the robber wins. We prove that the robber wins on graphs with girth at most 5. We also improve the bounds on a problem 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 m, where m is the minimum of n and a quantity related to the "metric dimension" of G. We obtain smaller thresholds for complete bipartite graphs and grids.

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*) completes the round by scanning a vertex in the graph and learning the distance from that vertex to the location of the robber. If the cop can then determine that location, the cop wins. If the robber can avoid this forever, then the robber wins. Winning does not require the cop to scan the vertex where the robber is.

Our model is like that of Seager [12], but her model also forbids the robber from moving to the vertex scanned in the previous round. We call this the *no-backtrack condition*; it makes the cop stronger. Hence our sufficient conditions for the cop to win are also sufficient in Seager's model. In Section 5 we provide some sufficient conditions for the robber to win; these do not apply to Seager's model.

Pursuit games on graphs were introduced by Parsons in [11] and have been studied extensively under various names (searching, sweeping, clearing, hunter/rabbit, etc.). Parsons' original problem involves one omniscient evader and pursuers who learn nothing about the evader's location except by occupying the same vertex as the evader. A variant where the cop and robber know each other's locations and the cop must land on the robber to win is studied in [10]. Another variant where the robber can move any distance along paths without cops is related to treewidth [13]. A randomized algorithm for cops and robbers with limited visibility is described in [6]. Surveys describing other variants include [1,3].

In contrast to these models, in our problem the cop is never actually on the graph. The cop may scan any vertex, remotely. Winning means only that the cop has determined the vertex where the robber must be, not that the cop has captured the robber. Models involving distance information could arise in the setting of sensors in a network (see [8]).

[*] Corresponding author. Tel.: +1 1217 333 1863.
E-mail addresses: s-jcarrah1@math.unl.edu (J. Carraher), ichoi4@illinois.edu (I. Choi), delcour2@illinois.edu (M. Delcourt), lericks4@uiuc.edu (L.H. Erickson), west@math.uiuc.edu (D.B. West).

0304-3975/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2012.06.035

Seager [12] proved that under the no-backtrack condition, the cop wins on cycles other than the 5-cycle and on 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 also considered whether for any graph G, the cop wins on some subdivision of G. A referee of our paper provided an easy argument that we include here as Proposition 1.1.

A *thread* in a graph is a path whose internal vertices have degree 2. The operation of *m-subdivision* replaces each edge in a graph with a thread of length m through new vertices; the notation for the resulting graph is $G^{1/m}$. (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.) For convenience in this argument, we temporarily use "branch vertex" to mean a vertex of $G^{1/m}$ that corresponds directly to a vertex of G.

Proposition 1.1. *For any graph G with n vertices, the cop wins on $G^{1/(3n)}$.*

Proof. Scanning the branch vertices in turn eventually yields a branch vertex v from which the distance to the robber is less than $2n$ at that moment. During the next n rounds the robber remains at distance less than $3n$ from v. Scanning the branch vertices corresponding to neighbors of v in G will locate the robber. If a scan at u returns distance less than $3n$, then the robber is located within the thread from u to v. Moving to another one of these threads requires the robber to pass through v, at which point the distance from any neighboring branch vertex will be exactly $3n$ and the robber will be located at v. □

This argument raises the natural problem of reducing the bound on m in the m-subdivision where the cop wins. We improve the general bound and provide even smaller bounds for some special graphs. The argument of Proposition 1.1 motivates our strategy for the cop. Scanning a special set finds candidates for a branch vertex z such that the robber stays within distance m of z during the first m rounds. Next, scanning the candidates produces a branch vertex within distance m of the robber. Finally, the last (repeatable) phase locates the robber or selects another branch vertex that the robber is now closer to. Care is needed because using shorter threads may allow the robber to escape from the union of the threads incident to z before being located.

We prove that if m exceeds both the maximum degree $\Delta(G)$ and a quantity involving the number of vertices of G that can arise as candidates, then the cop wins on $G^{1/m}$. Thus the cop wins when $m = n$, but in general the bound is smaller. For $W = \{w_1, \ldots, w_k\} \subseteq V(G)$ and $v \in V(G)$, the distances in G from v to $w_1, \ldots, w_k$ form a k-tuple $(d_G(v, w_1), \ldots, d_G(v, w_k))$. 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. Our main result is that if G has n vertices, then the cop wins on $G^{1/m}$ when $m > \min\{n - 1, \max\{\mu(G) + 2^{\mu(G)}, \Delta(G)\}\}$.

The metric dimension parameter was introduced independently by Slater [14] and by Harary and Melter [4]. Computing it is NP-complete [7]. 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].

The relevance of metric dimension to this problem 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 $G^{1/m}$.

We prove our main result in Section 2. Sections 3 and 4 reduce the threshold on m for the cop to win in $G^{1/m}$ when G is a complete bipartite graph or a grid; for a grid $m \geq 2$ suffices. Section 5 shows that the robber wins on graphs having a cycle of length at most 5.

Several natural questions remain open. First, we conjecture that our main result is sharp in the following sense.

Conjecture 1.2. *For $n \geq 4$, the cop wins the game on $K_n^{1/m}$ if and only if $m \geq n$.*

Our results depend heavily on subdividing edges into paths of equal length. Nevertheless, we conjecture that this restriction is not necessary.

Conjecture 1.3. *If the cop wins the game on a graph G, then subdividing one edge of G produces another graph on which the cop wins.*

Also, although the robber wins on the 6-cycle itself, it remains open whether the robber wins on all graphs whose shortest cycles have length 6.

2. Cop-win subdivisions of general graphs

Since the cop wins on a path by scanning one endpoint, 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 $H = G^{1/m}$, let $\hat{V}$ be the set of vertices in H that correspond to vertices of G. View $\hat{V}$ as $V(H) \cap V(G)$, so that these vertices of H can also be discussed as vertices of G. For $v \in \hat{V}$, let the *span* of v, denoted $\mathrm{Span}(v)$, 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. Let $\sigma(q)$ denote the result of scanning a vertex $q \in V(H)$ (the distance from q to the robber at that time). Let W denote a fixed resolving set of G, with vertices $w_1, \ldots, w_k$, and let $\hat{W} = W \cap \hat{V}$. For $v \in V(G) = \hat{V}$, the *W-vector* of v is the k-tuple whose ith 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 $v \in \hat{V}$.*

Proof. Let S be the set of positions occupied by the robber during some m consecutive rounds. If $v \in S$ for some $v \in \hat{V}$, then $S \subseteq \mathrm{Span}(v)$, since the robber cannot travel distance more than $m - 1$ between the first and last of these rounds.

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

We are given a resolving k-set W in G. When the vertices of $\hat{W}$ are scanned consecutively, we obtain a k-tuple D of distances from vertices of $\hat{W}$ to the robber at successive times. Since W is a resolving set in G, the W-vectors of vertices in G are distinct. If the robber remains at a vertex $z \in \hat{V}$ during this time, then $D = (md_G(z, w_1), \dots, md_G(z, w_k))$. Although the robber can move, by Lemma 2.2 the robber remains within the span of some vertex v in $\hat{V}$.

Lemma 2.3. *Let D be the k-tuple of distances to the robber returned by scanning the vertices of $\hat{W}$ in succession. If the robber stays in $\mathrm{Span}(v)$ during this time, where $v \in \hat{V}$, then*

$$D = (md_G(v, w_1) + \rho_1, \dots, md_G(v, w_k) + \rho_k)$$

for constants $\rho_1, \dots, \rho_k$ satisfying $-(m - 1) \le \rho_i \le m - 1$.

Proof. If $z \in \mathrm{Span}(v)$, then $md_G(v, w_i) - (m - 1) \le d_H(z, w_i) \le md_G(v, w_i) + (m - 1)$, by the triangle inequality. □

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

Lemma 2.4. *If the distances in G from vertices v_1 and v_2 to some vertex of W differ by at least 2, then v_1 and v_2 are not both candidates.*

Proof. Name v_1 and v_2 so that $d_G(w, v_1) < d_G(w, v_2)$. Since $d_G(w, v_2) - d_G(w, v_1) \ge 2$ it follows that $d_H(w, v_2) \ge d_H(w, v_1) + 2m$. Let x be the position of the robber when w is scanned. If both v_1 and v_2 are candidates, then $d_H(w, v_2) - (m - 1) \le d_H(w, x) \le d_H(w, v_1) + (m - 1)$, a contradiction. □

Corollary 2.5. *There are at most 2^k candidates.*

Proof. When v is a candidate, the value in position i of D is $md_G(w_i, v) + \rho_i$ with $-(m - 1) \le \rho_i \le m - 1$. An integer is within $m - 1$ of at most two multiples of m. 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^k resulting possible W-vectors for candidates determine at most one vertex each. □

We present an algorithm to locate the robber in $G^{1/m}$ when m is sufficiently large. As suggested in the introduction, the algorithm has several phases. The first phase, ScanW, finds a set of candidates. The second phase, PickOne, selects a candidate z whose span contains the robber when selected. The third phase, GetCloser, scans the neighbors of z in G until it locates the robber or finds another vertex z' in $\hat{V}$ that is closer to the robber than z was when it was selected. This process is repeated fewer than m times and can only end by locating the robber. Note that while we are scanning neighbors of z in GetCloser, the robber may leave $\mathrm{Span}(z)$.

Theorem 2.6. *Let $m_0 = \min\{n - 1, \max\{\mu(G) + 2^{\mu(G)}, \Delta(G)\}\}$, where $n = |V(G)|$. For $H = G^{1/m}$ with $m > m_0$, the cop can locate the robber in H within $m(\Delta(G) + 1)$ rounds.*

Proof. We provide a strategy for the cop to win. Let W be a resolving set in G of size $\mu(G)$.

The first phase, ScanW, scans all of $\hat{W}$ 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. Corollary 2.5 yields $|C| \le 2^{\mu(G)}$. However, if $\mu(G) + 2^{\mu(G)} > n$, then we skip this phase and just let $C = \hat{V}$.

Next, PickOne scans vertices of C until a vertex $z \in C$ is scanned such that $\sigma(z) < m$; we show that this must occur. Since m is at least the number of rounds in ScanW and PickOne together, Lemma 2.2 implies that the robber remains in the span of one vertex of $\hat{V}$ (call it z) from the beginning of the ScanW phase until all of C (if needed) is scanned. By Lemma 2.3, this vertex z is in C. When z is scanned, by the definition of "span" we obtain $\sigma(z) < m$. Let ρ be the value of $\sigma(z)$ at that time; at this point we end PickOne.

We now enter GetCloser with parameters z and ρ. Let $T = \{v \in \hat{V} \colon d_H(v, z) = m\}$. Note that $|T| \le \Delta(G)$. In GetCloser(z, ρ), 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, we restart GetCloser with parameters z' and $\sigma(z')$. Each trip through GetCloser scans at most $\Delta(G)$ vertices; we must show that it locates the robber or ends with another vertex of $\hat{V}$ closer to the robber.

Let $M(z) = \{x \in V(H) \colon \lfloor m/2 \rfloor \le d_H(z, x) \le \lceil m/2 \rceil\}$; we call $M(z)$ the *midway set* of z. Phase GetCloser(z, ρ) sets the *transition flag* (immediately) if at some point the scan value is a multiple of m. Phase GetCloser(z, ρ) sets the *confusion flag* if at some point the scan value is congruent to $m/2$ modulo m (when m is even) or if two consecutive scan values are congruent to $(m\pm 1)/2$ modulo m (when m is odd). This criterion is considered first when GetCloser(z, ρ) is entered, with scan value ρ (thus in the odd case the last scan value in the previous phase may also be examined). We consider several cases depending on these flags and the scan value for the current vertex $v \in T$.

Case 1: *$\sigma(v) < m$ and the transition flag is not set.* Since $\sigma(v) < m$, the robber is in Span(v). (Note that $\sigma(v) \neq 0$, since otherwise the transition flag would be set.) Since the transition flag is not set, the robber has not visited $\hat{V}$ in this phase, so the robber is in the z, v-thread. Therefore, $\sigma(v)$ determines the robber's location.

Case 2: *$\sigma(v) < \rho$ and the transition flag is set.* If $\sigma(v) = 0$, then the robber is at v. Otherwise, unset the flags and start GetCloser$(v, \sigma(v))$.

Case 3: *$\sigma(v) = m > 2\rho$ and the confusion flag is not set.* Since $\sigma(v) = m$, the robber is in T or at z. However, since $\rho < m/2$ and the confusion flag is not set, the robber started between z and the midway set (inclusive) on its thread and has not left that portion of the thread. Therefore, the robber cannot be in T and is at z.

Case 4: *v is not the last vertex of T to be scanned, and no earlier case now applies.* Set the transition or confusion flag if the relevant condition holds, and continue.

Case 5: *v is the last vertex of T to be scanned, and no earlier case now applies.* If this case is reached, then $\sigma(v) \geq \rho$ and the transition flag has been set by entering z (we explain this at further length below). If $\rho \leq \sigma(v) \leq m$, then the robber is in the thread joining z and v and is located. If $\sigma(v) > m$, then the robber is at distance $\sigma(v) - m$ from z. Furthermore, $\sigma(v) - m \leq \rho - 1$. Unset the flags and start GetCloser$(z, \sigma(v) - m)$.

It remains to justify that GetCloser(z, ρ) operates as claimed. If the transition flag is never set, then the robber never leaves the original thread, and Case 1 applies when the endpoint of that thread other than z is scanned.

If the transition flag is set by the robber entering T, then at least $m - \rho$ rounds occur by the time the robber enters T, say at vertex u. If u was scanned before the robber reached it, then Case 1 applied at that time, and the transition flag in fact was not set. Hence u is scanned after the robber reaches it. Since $|T| < m$, fewer than ρ rounds occur in the phase after the robber reaches u. Hence $\sigma(u) < \rho$ when u is scanned, and Case 2 applies.

If the transition flag is set by the robber visiting z without first setting the confusion flag, then $\sigma(v) = m$ when the transition flag is set. Since also the midway set was not traversed (otherwise the confusion flag would have been set), $\rho < m/2$, and hence Case 3 applies.

In the remaining case, the transition flag is set by the robber visiting z after setting the confusion flag. The robber has not entered T during this phase, since it takes at least m rounds to move from T to z. Since the confusion flag has been set, the robber has visited $M(z)$ (if m is even), or ended two consecutive rounds in $M(z)$ (if m is odd) (see Fig. 1).

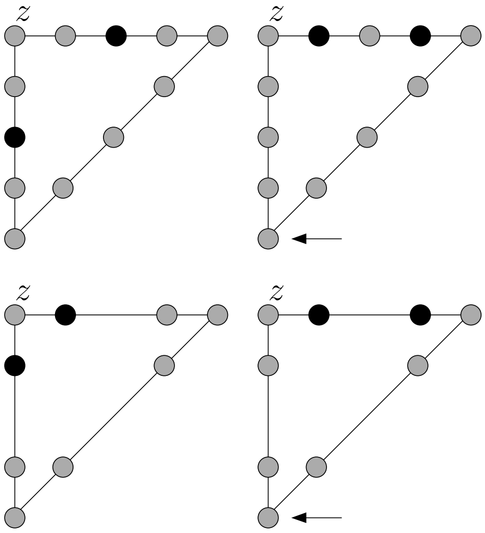

Fig. 1. Scanning a vertex after the confusion flag is set.

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.

When m is odd, 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 two consecutive rounds in $M(z)$ (one of which may be in the previous phase). 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.

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 $m - 1$ at the start of the first GetCloser phase, this phase occurs at most $m - 1$ times. Each such phase takes at most $\Delta(G)$ rounds. The ScanW and PickOne phases together take at most m rounds. Therefore, the entire algorithm takes at most $m(\Delta(G)+1)$ rounds to find the robber. □

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 upper bound n in Theorem 2.6 is better than the upper bound $1+\max\{\mu(G)+2^{\mu(G)}, \Delta(G)\}$ when $\mu(G) > \log_2 n$.

A bound in terms of $\mu(G)$ alone, weaker than Theorem 2.6, is obtained by replacing $\Delta(G)$ with $3^{\mu(G)}$. When u is a neighbor of v, and w belongs to a smallest resolving set W, the distances of u and v from w differ by at most 1. Since W is a resolving set, there are fewer than $3^{\mu(G)}$ distance vectors of neighbors of v, with no duplication, so $d(v) < 3^{\mu(G)}$. Hernando et al. [5] noted more generally that within distance k of a vertex in G there are at most $k(2k+1)^{\mu(G)}$ vertices.

3. Subdivisions of complete bipartite graphs

When G is a complete bipartite graph, we can further reduce the threshold on m for the cop to win in $G^{1/m}$. By examining the two partite sets in separate phases, we reduce the threshold on m for $K_{a,b}$ to $\max\{a, b\}$ (or to $a+1$ if $a=b$).

The overall strategy is like that of Theorem 2.6. An initial phase finds a vertex of the small part within distance m of the robber. Phases that scan the two parts then alternate. Like GetCloser(z, ρ), these phases have two arguments, the last scanned vertex in the previous phase (in the other partite set) and the scan distance at that time. Again a phase seeks to locate the robber or reduce the scan distance that starts the next phase. However, when $m=\max\{a, b\}$ we are only guaranteed to reduce the scan distance after two phases.

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

Proof. Let $G=K_{a,b}$, having partite sets X_0 and X_1 with $|X_0|=a$ and $|X_1|=b$. We present a cop strategy. In the first phase, called BiPickOne, we scan vertices of X_1 until we scan some vertex $z \in X_1$ such that $\sigma(z) \leq m$. If the robber visits X_0, then BiPickOne ends on that round, which returns scan value m. If the robber does not visit X_0, then the robber remains in the span of one vertex $z \in X_1$ throughout the phase, and $\sigma(z) < m$ when z is scanned. Let ρ be the scan distance when BiPickOne ends.

We next enter Scan$_0(z, \rho)$. We alternate two types of phases, Scan$_0$ and Scan$_1$, where Scan$_i$ scans vertices of X_i until the robber is located or we are able to start Scan$_{1-i}$ with a smaller scan value. The two types are almost the same, so we describe them together.

Since $|X_i| \leq m$, Scan$_i$ takes at most m rounds (with strict inequality if $i=1$). Note that the distance in H between two vertices of the same partite set of G is $2m$, while the distance between vertices of distinct partite sets is m. No distance exceeds $2m$. Say that the *transition flag* is set if the scan distance was $2m$ at some previous round in this phase. Let v denote the currently scanned vertex (in X_i) during Scan$_i(z, \rho)$. We consider several cases, depending on $\sigma(v)$ and the transition flag.

Case 1: *$\sigma(v) < m$ and the transition flag is not set.* Since $\sigma(v) < m$, the robber is in Span(v). Since the transition flag is not set, the robber has not visited X_i in this phase, so the robber remains in Span(z). Therefore, the robber is in the z, v-thread, and $\sigma(v)$ determines the robber's location.

Case 2: *$\sigma(v) \leq \rho - i$.* If $\sigma(v)=0$, then the robber is at v. Otherwise, unset the flag and start Scan$_{1-i}(v, \sigma(v))$. Note that $\sigma(v) \leq \rho$ if $i=0$, while $\sigma(v) < \rho$ if $i=1$.

Case 3: *$\sigma(v)=m$.* The robber is now in X_{1-i}. If $\rho < m$ or $i=1$, then since there are not enough rounds in Scan$_i(z, \rho)$ for the robber to reach any vertex of X_{1-i} other than z, the robber is located. If $\rho=m=a$ and $i=0$, then unset the flag and start Scan$_1(v, m)$.

Case 4: *No case above applies.* Continue, setting the transition flag if $\sigma(v)=2m$.

If Case 3 does not occur, then the robber does not visit z during the phase. The robber then remains in the original thread and produces Case 1 unless the transition flag is set by entering X_i at some vertex z'. Since the robber begins the phase ρ steps from z, it takes $m-\rho$ steps to reach z'. If z' is scanned before the robber reaches it, then Case 1 occurs. After the robber reaches z', at most ρ steps remain in Scan$_i(z, \rho)$, with strict inequality when $i=1$. Hence when z' is scanned, Case 2 occurs.

The exception in Case 3 can only occur during the first instance of Scan$_0$. The subsequent first instance of Scan$_1$ locates the robber or ends with a scan value less than m, since $b < m$.

The BiPickOne phase takes at most b rounds and results in $\rho \leq m$. Each pair of calls to Scan$_0$ and Scan$_1$ that does not locate the robber decreases ρ by at least 1 and scans each vertex at most once. Hence the cop locates the robber in at most $b+m(a+b)$ rounds. □

If actually $m > a$ in Theorem 3.1, then the proof is slightly simpler, with the condition $\sigma(v) < \rho$ in both Scan$_0$ and Scan$_1$ in Case 2. Now the scan value strictly decreases with each phase, and the robber is located within $b+m(a+b)/2$ rounds.

4. Subdivisions of grids

The *grid* $G_{k,l}$ is the graph with vertex set $\{v_{i,j}\colon 1 \le i \le k, 1 \le j \le l\}$ defined by making $v_{i,j}$ and $v_{i',j'}$ adjacent if and only if $|i-i'|+|j-j'|=1$ (see Fig. 2). When $k, l \ge 2$, the graph $G_{k,l}$ contains 4-cycles. We will show (in Section 5) that the robber wins on such graphs. A very special case of various results in [7,9] is that grids have metric dimension 2 ($\{v_{1,1}, v_{k,1}\}$ is a resolving set); Theorem 2.6 thus implies that the cop wins on $G_{k,l}^{1/m}$ when $m \ge 6$. Here we lower the threshold to 2.

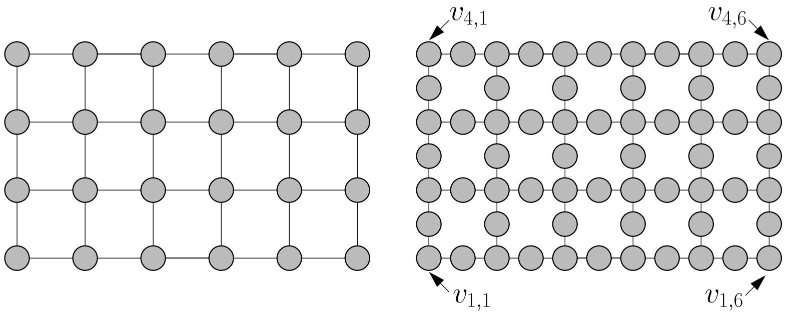

Fig. 2. The graph $G_{4,6}$ and the graph $G_{4,6}^{1/2}$.

Theorem 4.1. *If $H = G_{k,l}^{1/m}$ with $m \ge 2$, 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 $\hat{V}$, at a vertex in $\{v_{i,j}\colon i+j-2=\sigma(v_{1,1})/m\}$. Call this set S (Fig. 3 top left). After moving in round 2, the robber is in the set S' consisting of all vertices within distance 1 of 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 (Fig. 3 top right).

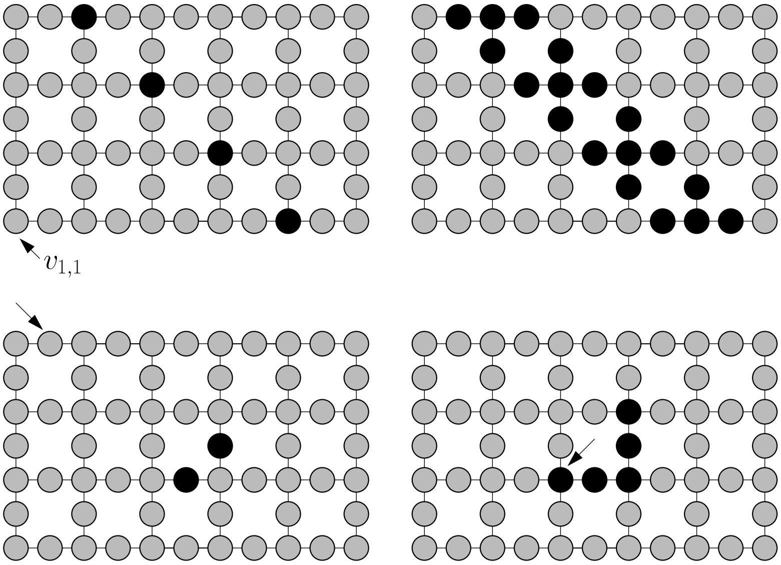

Fig. 3. The game on the grid when $\sigma(v_{1,1})$ is a multiple of m.

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

If the set has size 2, then the robber moves in round 3. The robber can remain still, move back to its original location in $\hat{V}$, or move one step farther away. 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 (Fig. 3 bottom right).

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 (Fig. 4 left).

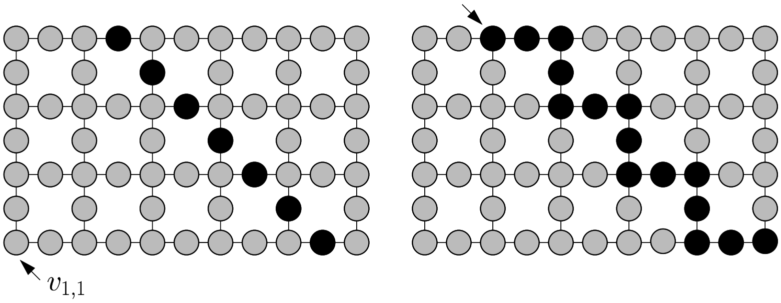

Fig. 4. The game on the grid when $\sigma(v_{1,1})$ is not a multiple of m.

In round 2, the robber moves at most one 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 (Fig. 4 right). Again distances along the path are distances in H, so the cop locates the robber by scanning an endpoint of this path. □

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 [12]. However, the robber wins on graphs having small cycles. Here we must clarify the meaning of "robber wins". The robber can never guarantee not being located, since the cop may get lucky and scan the vertex where the robber is. Hence the meaning of "robber wins" is that the cop cannot guarantee winning.

When π and τ are partitions of a set S, we say that π is a *refinement* of τ if every member of π is contained in one member of τ.

Theorem 5.1. *The robber wins on any graph G having a cycle of length at most* 5.

Proof. Let C be a shortest cycle in G. The robber generously agrees to stay within $V(C)$. Even with this restriction on the robber, 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 confined to one such set. If the partition from scanning a vertex v' is a refinement of the partition given by scanning v, then the cop gains more information by scanning v'. Since C is an induced subgraph, when C has odd length the most refined such partitions put one vertex u by itself and pair the vertices of C having equal distance along C from u (see Fig. 5). Such a partition can arise by scanning u, for example. When C has length 4, it may be possible to have two singleton classes and one of size 2, or to have two classes of size 2 consisting of consecutive pairs (see Fig. 5). We make the cop strongest by allowing all these partitions of C to be available by scanning appropriate vertices.

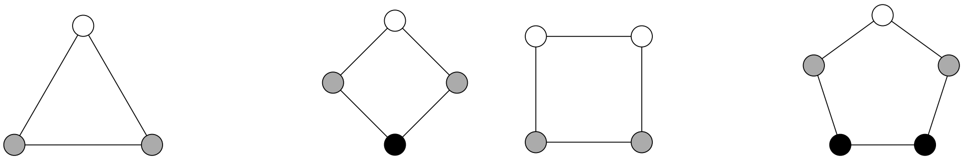

Fig. 5. The non-refinable partitions of small cycles, up to automorphism.

At the initial scan, the robber may be anywhere on C, so we may assume that the robber is in a class of size 2 in the partition chosen by the cop. We show that the robber can maintain the property of possibly being at either vertex of some class of size 2 in the partition chosen by the cop. Since the cop does not know which vertex of the current pair contains the robber, to maintain this property after the next round it suffices that both vertices of some class of size 2 in the next partition are reachable from the current class.

When C is a 3-cycle, each partition has only one class of size 2 (call it the "pair"), and each vertex of the pair in the next partition is equal to or adjacent to a vertex of the pair in the current partition.

When C is a 4-cycle, the cop can choose whether the next class of size 2 consists of adjacent or nonadjacent vertices. Nevertheless, each vertex in any pair in $V(C)$ is adjacent to or belongs to each other pair in $V(C)$, so again the robber can maintain the desired property.

When C is a 5-cycle, the initial scan may have the robber in the class with nonadjacent vertices. The robber may always be at either vertex of the nonadjacent class in the next partition, since each vertex in that pair has a neighbor in (or belongs to) each pair of nonadjacent vertices. □

Seager [12] 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 particular, when G is a 6-cycle in our model, the robber can always choose a class of size 2 whose vertices are both reachable from the current class of size 2. However, we have not been able to determine who wins in our model on every graph whose shortest cycles have length 6, since vertices outside the 6-cycle can produce other partitions. The cop can then prevent the robber from maintaining the property in Theorem 5.1. The robber would need a more complex strategy that involves leaving the 6-cycle.

Acknowledgments

The first author was partially supported by NSF grant DMS 0838434 EMSW21MCTP: Research Experience for Graduate Students and NSF grant DMS 0914815. The third author was partially supported by NSF grant DMS 0838434 EMSW21MCTP: Research Experience for Graduate Students and NSF Graduate Research Fellowship DGE 1144245. The fourth author was 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. The fifth author was partially supported by NSA grant H98230-10-1-0363.

References

[1] B. Alspach, Searching and sweeping graphs: a brief survey, Matematiche 59 (2004) 5–37.
[2] G. Chartrand, L. Eroh, M.A. Johnson, O.R. Oellermann, Resolvability in graphs and the metric dimension of a graph, Discrete Applied Mathematics 105 (2000) 99–113.
[3] F.V. Fomin, D.M. Thilikos, An annotated bibliography on guaranteed graph searching, Theoret. Comput. Sci. 399 (2008) 236–245.
[4] F. Harary, R.A. Melter, On the metric dimension of a graph, Ars Combinatorica 2 (1976) 191–195.
[5] C. Hernando, M. Mora, P.J. Slater, D.R. Wood, Fault-tolerant metric dimension of graphs, in: Convexity in Discrete Structures, in: Ramanujan Math. Soc. Lect. Notes Ser., 5, Ramanujan Math. Soc., 2008, pp. 81–85.
[6] V. Isler, S. Kannan, S. Khanna, Randomized pursuit-evasion with local visibility, SIAM J. Discrete Math. 20 (2006) 26–41.
[7] S. Khuller, B. Raghavachari, A. Rosenfeld, Landmarks in graphs, Discrete Applied Mathematics 70 (1996) 217–229.
[8] S.M. LaValle, Sensing and Filtering: A Fresh Perspective Based on Preimages and Information Spaces, in: Foundations and Trends in Robotics Series, vol. 1 (4), Now Publishers, Delft, The Netherlands, 2012.
[9] R.A. Melter, I. Tomescu, Metric bases in digital geometry, Computer Vision, Graphics, and Image Processing 25 (1984) 113–121.
[10] R. Nowakoski, P. Winkler, Vertex-to-vertex pursuit in a graph, Discrete Math. 43 (1983) 235–239.
[11] T. Parsons, Pursuit-evasion in a graph, in: Y. Alavi, D. Lick (Eds.), Theory and Applications of Graphs, in: Lecture Notes in Mathematics, vol. 642, Springer, 1978, pp. 426–441.
[12] S. Seager, Locating a robber on a graph, Discrete Math., to appear.
[13] P.D. Seymour, R. Thomas, Graph searching and a min–max theorem for tree-width, J. Combinatorial Theory (B) 58 (1993) 22–33.
[14] P.J. Slater, Leaves of trees, in: Proc. Sixth Southeastern Conf. Combin., Graph Theory, Computing, in: Congressus Numer., 14, 1975, pp. 549–559.