

Approximately locating an invisible agent in a graph with relative distance queries

Dennis Dayanikli

Dieter Rautenbach

Institut für Optimierung und Operations Research, Universität Ulm, Ulm, Germany,
{dennis.dayanikli,dieter.rautenbach}@uni-ulm.de

Abstract

In a pursuit evasion game on a finite, simple, undirected, and connected graph G , a first player visits vertices m_1, m_2, \dots of G , where m_{i+1} is in the closed neighborhood of m_i for every i , and a second player probes arbitrary vertices c_1, c_2, \dots of G , and learns whether or not the distance between c_{i+1} and m_{i+1} is at most the distance between c_i and m_i . Up to what distance d can the second player determine the position of the first? For trees of bounded maximum degree and grids, we show that d is bounded by a constant. We conjecture that $d = O(\log n)$ for every graph G of order n , and show that $d = 0$ if m_{i+1} may differ from m_i only if i is a multiple of some sufficiently large integer.

Keywords: pursuit and evasion game

1 Introduction

We study a variant of pursuit and evasion games formalized and studied by Britnell and Wildon [2], Komarov and Winkler [6], Haslegrave [4], Seager [8], and Rautenbach and Schneider [7]. In these games, further studied in [1, 3, 5, 9], one player tries to catch or locate a second player moving along the edges of a graph, using information concerning the current position of the second player. Our game is also played by two players on a finite, simple, undirected, and connected graph G known to both of them, and proceeds in discrete time steps numbered by positive integers. One player, called the *mouse*, moves along the edges of G . At time i , the mouse occupies some vertex m_i of G , and, if i is at least 2, then m_i is either m_{i-1} or a neighbor of m_{i-1} , that is, m_i belongs to the closed neighborhood $N_G[m_{i-1}]$ of m_{i-1} in G , and the mouse can be considered to move with unit speed. The second player, called the *cat*, probes vertices of G one by one in the same discrete time steps. At time i , the cat probes some vertex c_i of G , where c_i can be chosen without any restriction within the vertex set $V(G)$ of G .

The essential difference of our game, as compared to those mentioned above, consists in the information provided to the cat. If i is at least 2, then, after m_i and c_i have been decided by the two players, the cat learns whether

- $d_i \leq d_{i-1}$ or
- $d_i > d_{i-1}$,

where d_i denotes the distance $\text{dist}_G(c_i, m_i)$ in G between c_i and m_i . The goal of the cat is to locate the mouse as precisely as possible, while the goal of the mouse is to hinder being well located. To make this more precise, we introduce some further terminology.

A *game* g on G is a pair of sequences $((m_i)_{i \in \mathbb{N}}, (c_i)_{i \in \mathbb{N}})$ of possible moves m_i for the mouse, and c_i for the cat. For such a game g , and an integer i at least 2, let

$$b_i = \begin{cases} 1 & , \text{ if } d_i \leq d_{i-1}, \text{ and} \\ 0 & , \text{ if } d_i > d_{i-1}, \end{cases}$$

that is, the information available to the cat for its choice of c_{i+1} consists of G and the $i - 1$ bits b_2, \dots, b_i . Note that the cat chooses c_1 and c_2 without any information about the whereabouts of the mouse. Based on the available information, the cat knows that m_i belongs to the set M_i , where M_i is the set of all vertices u of G such that there are vertices $\tilde{m}_1, \dots, \tilde{m}_i$ of G with

- $\tilde{m}_i = u$,
- $\tilde{m}_j \in N_G[\tilde{m}_{j-1}]$ for every $j \in [i] \setminus \{1\}$, and
- $\text{dist}_G(c_j, \tilde{m}_j) \leq \text{dist}_G(c_{j-1}, \tilde{m}_{j-1})$ if and only if $b_j = 1$ for every $j \in [i] \setminus \{1\}$,

where $[i]$ denotes the set of positive integers at most i .

If the *radius* $\text{rad}_G(M)$ of a set M of vertices of G is defined as

$$\min \left\{ \max \left\{ \text{dist}_G(u, m) : m \in M \right\} : u \in V(G) \right\},$$

then the cat wants to minimize the radius of M_i . Note that the vertex u in this definition may not belong to M .

We say that the cat follows a *strategy* $(c_1, c_2; f)$ in the game $((m_i)_{i \in \mathbb{N}}, (c_i)_{i \in \mathbb{N}})$ on G if c_1 and c_2 are vertices of G , and

$$f : \bigcup_{i \in \mathbb{N}} \{0, 1\}^i \rightarrow V(G)$$

is a function such that c_1 and c_2 are the two first vertices probed by the cat, and $c_{i+1} = f(b_2, \dots, b_i)$ for every integer i at least 2. Furthermore, we say that *the cat can localize the mouse up to distance d within time t on G* if there is some strategy σ such that for every game $((m_i)_{i \in \mathbb{N}}, (c_i)_{i \in \mathbb{N}})$ on G in which the cat follows the strategy σ , there is some positive integer i at most t with $\text{rad}_G(M_i) \leq d$.

While the cat only knows G and the b_i , and therefore also the set M_i , we may assume that the mouse knows G and also any strategy followed by the cat. Note that we consider

a game to be infinite, and that we did not specify any winning conditions. A reasonable way to do so is to fix a distance threshold d , and to declare the cat to be the winner on the pair (G, d) if it can locate the mouse up to distance d within finite time.

A natural question concerning our game is how precisely the cat can localize the mouse on a given graph. Our first result provides an answer for trees of bounded maximum degree.

Theorem 1.1. *The cat can localize the mouse up to distance $4\Delta - 6$ within time $O(h\Delta)$ on every tree T of maximum degree Δ at least 2 and radius h .*

Another natural type of graphs to consider are grids, that is, the Cartesian product $P_n \square P_m$ of paths. For these we show the following.

Theorem 1.2. *The cat can localize the mouse up to distance 8 within time $O(\log n)$ on the grid $P_n \square P_n$.*

Both our results concern graphs of bounded maximum degree, and we pose the following general conjecture.

Conjecture 1.3. *The cat can localize the mouse up to distance $O(\log n)$ on a connected graph G of order n .*

The reason for the $O(\log n)$ term in this conjecture is that this many bits suffice to identify each vertex, while the mouse may move this many units of distance in the time needed to acquire this many bits.

Our final result establishes a weak version of Conjecture 1.3. In order to facilitate the task for the cat, we slow down the mouse as follows. For some positive integer k , a game $((m_i)_{i \in \mathbb{N}}, (c_i)_{i \in \mathbb{N}})$ on a graph G is k -slow if $m_i = m_{i-1}$ for every integer i at least 2 such that $(i-1) \not\equiv 0 \pmod k$, that is, the mouse can be considered to move with speed $1/k$. We say that *the cat can localize a k -slow mouse up to distance d on G* if there is some strategy σ such that for every k -slow game $((m_i)_{i \in \mathbb{N}}, (c_i)_{i \in \mathbb{N}})$ on G in which the cat follows the strategy σ , there is some positive integer i with $\text{rad}_G(M_i) \leq d$.

Theorem 1.4. *If Δ is an integer at least 2, then the cat can localize a 4Δ -slow mouse up to distance 0 on every connected graph G of maximum degree at most Δ .*

In Section 2 we prove our results, and in Section 3 we present some open problems.

2 Proofs

For all three of our results, we give simple proofs capturing essential observations. For Theorems 1.1 and 1.2, minor improvements are possible at the cost of tedious case analysis.

Proof of Theorem 1.1. Let T , Δ , and h be as in the statement. In order to express information gathered by the cat, we introduce the notation $u \xrightarrow{i} v$, where uv is an edge of T ,

and $i \in \mathbb{N}$, meaning that m_i belongs to the component of $T - uv$ that contains v .

Input: The relative distance information “ $d_i \leq d_{i-1}$ ” or “ $d_i > d_{i-1}$ ” after specifying c_i for some integer i at least 2.

Output: A statement of the form “ $\text{dist}_G(r, m_i) \leq 4\Delta - 6$ ” at the end of some round i .

```

1 begin
2    $r \leftarrow r_0; I \leftarrow \emptyset; i \leftarrow 1;$ 
3   while  $T_r$  has depth more than  $4\Delta - 6$  do
4     Let  $X$  be the set of children of  $r$ ;
5     while  $|X| \geq 2$  do
6       Let  $u$  and  $v$  be distinct vertices in  $X$ ;  $c_i \leftarrow u$ ;  $c_{i+1} \leftarrow v$ ;
7       if  $d_{i+1} \leq d_i$  then
8          $X \leftarrow X \setminus \{u\}; I \leftarrow I \cup \{u \xrightarrow{i+1} r\};$ 
9       else
10         $X \leftarrow X \setminus \{v\}; I \leftarrow I \cup \{v \xrightarrow{i+1} r\};$ 
11      end
12       $i \leftarrow i + 2;$ 
13    end
14    Let  $r^+$  be the unique element in  $X$ ;
15    Let  $Y$  be the set of children of  $r^+$ ;
16    replace  $\leftarrow$  false;
17    while  $\text{replace} = \text{false}$  and  $Y \neq \emptyset$  do
18      Let  $u$  be in  $Y$ ;  $c_i \leftarrow u$ ;  $c_{i+1} \leftarrow r$ ;
19      if  $d_{i+1} \leq d_i$  then
20         $Y \leftarrow Y \setminus \{u\}; I \leftarrow I \cup \{u \xrightarrow{i+1} r^+\};$ 
21      else
22        replace  $\leftarrow$  true;  $I \leftarrow I \cup \{r \xrightarrow{i+1} r^+\};$ 
23      end
24       $i \leftarrow i + 2;$ 
25    end
26    if  $\text{replace} = \text{true}$  then
27       $r \leftarrow r^+;$ 
28    else
29      return “ $\text{dist}_G(r, m_i) \leq 4\Delta - 6$ ”; break;
30    end
31  end
32  return “ $\text{dist}_G(r, m_i) \leq 4\Delta - 6$ ”;
33 end

```

Algorithm 1: CAT ON A TREE

A key ingredient for the cat's strategy is the following simple observation: If u and w are two vertices of T with a common neighbor v , and the cat chooses c_i equal to u and c_{i+1} equal to w for some $i \in \mathbb{N}$, then $d_{i+1} \leq d_i$ implies $u \xrightarrow{i+1} v$, while $d_{i+1} > d_i$ implies $w \xrightarrow{i+1} v$. In fact, if m_{i+1} does not belong to the component of $T - uv$ that contains v , then $d_{i+1} = \text{dist}_T(c_{i+1}, m_{i+1}) = \text{dist}_T(c_i, m_{i+1}) + 2 \geq \text{dist}_T(c_i, m_i) + 1 > d_i$, which proves the first implication, and the proof of the second implication is similar.

The cat chooses a center vertex r_0 , and considers T to be rooted in r_0 . By the choice of r_0 , the depth of T is h . For a vertex r of T , let T_r denote the subtree of T rooted in r that contains r and all descendants of r .

The algorithm CAT ON A TREE, cf. Algorithm 1, specifies the strategy for the cat. Note that the game progresses one round whenever the cat specifies some c_i . Throughout the game, the cat maintains a local root r initially equal to r_0 . The set I contains the information of the form “ $u \xrightarrow{i} v$ ” already gathered by the cat, and it is initially empty. In each iteration of the outer **while**-loop in line 3, the cat either replaces r with one of its children r^+ in line 27 or returns the statement “ $\text{dist}_G(r, m_i) \leq 4\Delta - 6$ ” in line 29 and terminates. At the beginning of each iteration of the outer **while**-loop, we have

$$\text{either } i = 1 \text{ or } \left(i > 1 \text{ and } m_{i-1} \text{ belongs to } T_r \right). \quad (1)$$

The second case is expressed by the element $r^- \xrightarrow{i-1} r$ of I added in line 22, where r^- is the parent of r . For the first iteration, (1) is trivial, and for later iterations, we will show it by an inductive argument. The outer **while**-loop is no longer performed if the depth of T_r is at most $4\Delta - 6$, which, by (1), implies that the output in line 32 is correct.

Now, we consider an iteration of the outer **while**-loop, and assume that (1) holds at its beginning. Let $i_0 + 1$ be the value of i at the beginning of that iteration, that is, $i_0 = 0$ for the very first iteration. The first inner **while**-loop in line 5 exploits the key observation made above. It follows that there is an ordering u_1, \dots, u_x of the children of r in T such that, at the end of the first inner **while**-loop,

$$u_j \xrightarrow{i_0+2j} r \in I \text{ for every } j \text{ in } [x-1], \quad (2)$$

and the vertex r^+ in line 14 equals u_x . Within the second inner **while**-loop in line 17, the cat may replace r with r^+ in line 27, in which case it concludes the current iteration of the outer **while**-loop. Note that this happens exactly if **replace** is set to **true**, and $r \xrightarrow{i+1} r^+$ is added to I in line 22, that is, (1) holds at the end of the considered iteration of the outer **while**-loop, which concludes the inductive proof of (1).

If **replace** is never set to **true** during the second inner **while**-loop, then, similarly as for the first inner **while**-loop, it follows that there is an ordering v_1, \dots, v_y of the children of r^+ in T such that at the end of the second inner **while**-loop, the set I contains

$$v_j \xrightarrow{i_0+2(x-1)+2j} r^+ \text{ for every } j \text{ in } [y]. \quad (3)$$

In this case, the cat returns the statement “ $\text{dist}_G(r, m_i) \leq 4\Delta - 6$ ” in line 29, where $i = i_0 + 2x + 2y - 2$, and terminates the computation. To show the correctness of CAT ON A TREE, we need to argue that this statement is correct. Indeed, if m_i belongs to T_{u_j} for some j in $[x - 1]$, then, by (2), $u_j \xrightarrow{i_0+2j} r \in I$, which implies that

$$\begin{aligned}
\text{dist}_T(r, m_i) &\leq i - (i_0 + 2j) \\
&= (i_0 + 2x + 2y - 2) - (i_0 + 2j) \\
&\stackrel{j \geq 1}{\leq} (i_0 + 2x + 2y - 2) - (i_0 + 2) \\
&= 2x + 2y - 4 \\
&\leq 2\Delta + 2(\Delta - 1) - 4 \\
&= 4\Delta - 6,
\end{aligned}$$

where we used $x \leq \Delta$ and $y \leq \Delta - 1$.

If m_i belongs to T_{v_j} for some j in $[y]$, then, by (3), $v_j \xrightarrow{i_0+2(x-1)+2j} r^+ \in I$, which implies that

$$\begin{aligned}
\text{dist}_T(r, m_i) &\leq \text{dist}_T(r, r^+) + i - (i_0 + 2x + 2j - 2) \\
&= 1 + (i_0 + 2x + 2y - 2) - (i_0 + 2x + 2j - 2) \\
&\stackrel{j \geq 1}{\leq} 1 + (i_0 + 2x + 2y - 2) - (i_0 + 2x) \\
&= 2y - 1 \\
&\leq 2\Delta - 3 \\
&\leq 4\Delta - 6,
\end{aligned}$$

where we used $y \leq \Delta - 1$.

If $m_i \in \{r, r^+\}$, then $\text{dist}_T(r, m_i) \leq 1 \leq 4\Delta - 6$, where we used $\Delta \geq 2$.

Finally, if m_i does not belong to T_r , then, by (1), $r^- \xrightarrow{i_0} r \in I$, which implies that

$$\begin{aligned}
\text{dist}_T(r, m_i) &\leq i - i_0 \\
&= (i_0 + 2x + 2y - 2) - i_0 \\
&= 2x + 2y - 2 \\
&\leq 2(\Delta - 1) + 2(\Delta - 1) - 2 \\
&= 4\Delta - 6,
\end{aligned}$$

where we used $x \leq \Delta - 1$ and $y \leq \Delta - 1$. Note that the existence of r^- implies that r has at most $\Delta - 1$ children. Altogether, the correctness of CAT ON A TREE follows, and we consider its running time.

The outer **while**-loop is executed at most $\max\{0, h - (4\Delta - 6)\}$ times. If, in some execution of the outer **while**-loop, r has x children, and r^+ has y children, then the first inner **while**-loop is executed $x - 1 \leq \Delta - 1$ times, while the second inner **while**-loop is

executed at most $y \leq \Delta - 1$ times. Since in each of these executions of the inner **while**-loops, the discrete time proceeds exactly two units, CAT ON A TREE terminates in some round i with $i \leq 2 \max\{0, h - (4\Delta - 6)\}((\Delta - 1) + (\Delta - 1)) = O(h\Delta)$, which completes the proof. \square

For $\Delta \geq 3$, with some more case analysis, it is possible to improve $4\Delta - 6$ to $4\Delta - 7$. For $\Delta = 2$, that is, on a path, say of order n , the cat can localize the mouse up to distance 2 within time $O(\log n)$. The corresponding strategy is similar to the strategy used in the following proof.

For integers i and j , let $[i, j]$ equal $\{k \in \mathbb{Z} : i \leq k \leq j\}$.

Proof of Theorem 1.2. Let $[n] \times [n]$ denote the vertex set of $P_n \square P_n$, where two vertices (x, y) and (x', y') are adjacent if and only if $|x - x'| + |y - y'| = 1$.

Suppose that

$$M_i \subseteq [x_i, x_i + \partial x_i] \times [y_i, y_i + \partial y_i] \subseteq [n] \times [n] \quad (4)$$

for some positive integers $i, x_i, \partial x_i, y_i$, and ∂y_i . For $i = 1$, for instance, $M_1 \subseteq [1, 1 + (n - 1)] \times [1, 1 + (n - 1)] = [n] \times [n]$. If ∂x_i is even, then let $p_x = \frac{\partial x_i}{2} - 1$, and if ∂x_i is odd, then let $p_x = \frac{\partial x_i}{2} - \frac{1}{2}$. Let p_y be defined similarly using ∂y_i . Now, the cat chooses

$$\begin{aligned} c_{i+1} &= (x_i + p_x, y_i + p_y), \\ c_{i+2} &= (x_i + p_x + 2, y_i + p_y), \text{ and} \\ c_{i+3} &= (x_i + p_x + 2, y_i + p_y + 2). \end{aligned}$$

Arguing similarly as for the key observation exploited in the proof of Theorem 1.1, it follows that

$$M_{i+3} \subseteq [x_{i+3}, x_{i+3} + \partial x_{i+3}] \times [y_{i+3}, y_{i+3} + \partial y_{i+3}] \subseteq [n] \times [n], \text{ where} \quad (5)$$

$$\partial x_{i+3} \leq \left\lceil \frac{\partial x_i}{2} \right\rceil + 4, \quad (6)$$

$$\partial y_{i+3} \leq \left\lceil \frac{\partial y_i}{2} \right\rceil + 3, \text{ and} \quad (7)$$

$x_{i+3}, \partial x_{i+3}, y_{i+3}$, and ∂y_{i+3} are positive integers. Indeed, suppose that ∂x_i is even, ∂y_i is odd, $d_{i+2} \leq d_{i+1}$, and $d_{i+3} > d_{i+2}$. In this case, it follows that

$$\begin{aligned} m_i &\in [x_i, x_i + \partial x_i] \times [y_i, y_i + \partial y_i], \\ m_{i+1} &\in [x_i - 1, x_i + \partial x_i + 1] \times [y_i - 1, y_i + \partial y_i + 1], \\ m_{i+2} &\in \left[x_i + \frac{\partial x_i}{2}, x_i + \partial x_i + 2\right] \times [y_i - 2, y_i + \partial y_i + 2], \text{ and} \\ m_{i+3} &\in \left[x_i + \frac{\partial x_i}{2} - 1, x_i + \partial x_i + 3\right] \times \left[y_i - 3, y_i + \frac{\partial y_i}{2} + \frac{1}{2}\right]. \end{aligned}$$

See Figure 1 for an illustration.

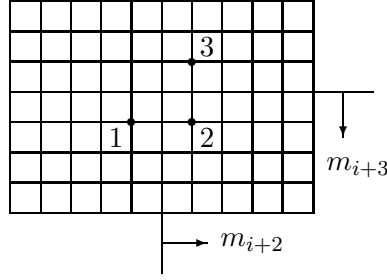


Figure 1: An illustration for the case $\partial x_i = 10$, $\partial y_i = 7$, $d_{i+2} \leq d_{i+1}$, and $d_{i+3} > d_{i+2}$.

We obtain

$$\begin{aligned}\partial x_{i+3} &= \left(x_i + \partial x_i + 3\right) - \left(x_i + \frac{\partial x_i}{2} - 1\right) = \frac{\partial x_i}{2} + 4 \text{ and} \\ \partial y_{i+3} &= \left(y_i + \frac{\partial y_i}{2} + \frac{1}{2}\right) - (y_i - 3) = \left\lceil \frac{\partial y_i}{2} \right\rceil + 3,\end{aligned}$$

and (5), (6), and (7) follow. All remaining cases are similar; we leave the details to the reader.

Note that $\lceil \frac{x}{2} \rceil + 4 < x$ for $x \geq 10$, and that $\lceil \frac{y}{2} \rceil + 3 < y$ for $y \geq 8$. Therefore, iteratively choosing the vertices as above, the cat can ensure that (4) holds with $\partial x_i \leq 9$ and $\partial y_i \leq 7$ for some $i = O(\log n)$. If either $\partial x_i < 9$ or $\partial y_i < 7$, then we obtain $\text{rad}_G(M_i) \leq 8$. If $\partial x_i = 9$ and $\partial y_i = 7$, then choosing c_{i+1} , c_{i+2} , and c_{i+3} with the roles of the x - and y -coordinates exchanged, we obtain, using (6) and (7), that (5) holds with $\partial x_{i+3} \leq \lceil \frac{9}{2} \rceil + 3 = 8$ and $\partial y_{i+3} \leq \lceil \frac{7}{2} \rceil + 4 = 8$, which implies $\text{rad}_G(M_{i+3}) \leq 8$. This completes the proof. \square

Proof of Theorem 1.4. We show that until the cat has not identified the position of the mouse in some round, for every positive integer i with $i \equiv 1 \pmod{4\Delta}$, the cat can choose a vertex r_i such that it can

- either identify the vertex $m_{i+4\Delta-1}$ in round $i + 4\Delta - 1$
- or choose a vertex $r_{i+4\Delta}$ with $\text{dist}_G(r_{i+4\Delta}, m_{i+4\Delta}) < \text{dist}_G(r_i, m_i)$,

which clearly implies the desired result.

Let r_1 be any vertex of G . Now, suppose that the positive integer i is such that $i \equiv 1 \pmod{4\Delta}$, and that r_i has been chosen. Note that $m_i = m_{i+1} = \dots = m_{i+4\Delta-1}$. The cat chooses

$$(c_i, c_{i+1}, \dots, c_{i+2d}) = (r_i, u_1, r_i, u_2, r_i, \dots, r_i, u_d, r_i),$$

where $N_G(r_i) = \{u_1, \dots, u_d\}$ for some $d \leq \Delta$.

If $\text{dist}_G(u_j, m_i) = \text{dist}_G(c_{i+2j-1}, m_i) > \text{dist}_G(c_{i+2j-2}, m_i) = \text{dist}_G(r_i, m_i)$ for every $j \in [d]$, then $m_{i+4\Delta-1} = m_i = r_i$, that is, the cat identifies $m_{i+4\Delta-1}$ in round $i + 4\Delta - 1$. Hence,

we may assume that $\text{dist}_G(r_i, m_i) = \text{dist}_G(c_{i+2j}, m_i) > \text{dist}_G(c_{i+2j-1}, m_i) = \text{dist}_G(u_j, m_i)$ for some $j \in [d]$. In this case, the cat chooses

$$(c_{i+2d+1}, c_{i+2d+2}, \dots, c_{i+2d+2d'+1}) = (u_j, v_1, u_j, v_2, u_j, \dots, u_j, v_{d'}, u_j),$$

where $N_G(u_j) = \{r_i, v_1, \dots, v_{d'}\}$ for some $d' \leq \Delta - 1$.

If $\text{dist}_G(v_\ell, m_i) = \text{dist}_G(c_{i+2d+2\ell}, m_i) > \text{dist}_G(c_{i+2d+2\ell-1}, m_i) = \text{dist}_G(u_j, m_i)$ for every $\ell \in [d']$, then $m_{i+4\Delta-1} = m_i = u_j$. Hence, we may assume that $\text{dist}_G(u_j, m_i) = \text{dist}_G(c_{i+2d+2\ell+1}, m_i) > \text{dist}_G(c_{i+2d+2\ell}, m_i) = \text{dist}_G(v_\ell, m_i)$ for some $\ell \in [d]$, which implies $\text{dist}_G(v_\ell, m_i) = \text{dist}_G(u_j, m_i) - 1 = \text{dist}_G(r_i, m_i) - 2$. In this case, the cat chooses $r_{i+4\Delta}$ equal to v_ℓ . Since $m_{i+4\Delta} \in N_G[m_{i+4\Delta-1}] = N_G[m_i]$, we obtain $\text{dist}_G(r_{i+4\Delta}, m_{i+4\Delta}) \leq \text{dist}_G(r_{i+4\Delta}, m_{i+4\Delta-1}) + 1 = \text{dist}_G(v_\ell, m_i) + 1 = \text{dist}_G(r_i, m_i) - 1$, which completes the proof. \square

3 Conclusion

We collect some problems for further research. It would be interesting to obtain tight bounds for the problems considered in Theorem 1.1 and Theorem 1.2. In this context, lower bounds on the minimum distance up to which the mouse can be located would be useful. An interesting special graph, and possible counterexample to Conjecture 1.3, seems to be the tree that arises by subdividing each edge of the star $K_{1,k}$ exactly $k - 1$ times. One may consider different slowness conditions such as, for instance, “ $(m_i \neq m_{i-1}) \wedge (m_j \neq m_{j-1}) \wedge (j > i) \Rightarrow j - i \geq k$ ” for some integer k corresponding to the inverse maximum speed of the mouse. For a proof of Conjecture 1.3, it might be useful that, for every graph G of maximum degree at most Δ , and every set M of vertices of G , there is an edge uv of G for which the two sets $\{m \in M : \text{dist}_G(u, m) > \text{dist}_G(v, m)\}$ and $\{m \in M : \text{dist}_G(u, m) < \text{dist}_G(v, m)\}$ both contain at least $\frac{|M|-1}{\Delta}$ vertices.

References

- [1] A. Brandt, J. Diemunsch, C. Erbes, J. LeGrand, C. Moffatt, A robber locating strategy for trees, *Discrete Applied Mathematics* 232 (2017) 99-106.
- [2] J.R. Britnell, M. Wildon, Finding a princess in a palace: A pursuit-evasion problem, *The Electronic Journal of Combinatorics* 20 (2013) #P25.
- [3] J. Carraher, I. Choi, M. Delcourt, L.H. Erickson, D.B. West, Locating a robber on a graph via distance queries, *Theoretical Computer Science* 463 (2012) 54-61.
- [4] J. Haslegrave, An evasion game on a graph, *Discrete Mathematics* 314 (2014) 1-5.
- [5] J. Haslegrave, R.A.B. Johnson, S. Koch, The Robber Locating game, *Discrete Mathematics* 339 (2016) 109-117.

- [6] N. Komarov, P. Winkler, Hunter & Mole, arXiv:1311.0211.
- [7] D. Rautenbach, M. Schneider, The Cat and the Noisy Mouse, to appear in Discrete Mathematics.
- [8] S. Seager, Locating a robber on a graph, Discrete Mathematics 312 (2012) 3265-3269.
- [9] S. Seager, Locating a backtracking robber on a tree, Theoretical Computer Science 539 (2014) 28-37.