Locating a robber on a graph

Suzanne Seager
Mount Saint Vincent University, Halifax, NS, Canada B3M 2J6

1. Introduction

Graph searching and graph locating both involve finding a specific vertex on a connected graph. Graph searching dates from Parsons [6] in 1976 and concerns capturing an evader hiding on a graph, where the parameters of interest are the minimum number of searchers required and the minimum guaranteed capture time. There are many variations; a good survey is Fomin and Thilikos [3]. One variation is the cop and robber game, initially posed independently by Nowakowski and Winkler [5] and Quilliot [7]. In this game, first the cop and then the robber choose a starting vertex on a connected graph. Then they take turns, at each turn either moving to an adjacent vertex or staying put. The cop's goal is to capture the robber by moving to the robber's vertex and the robber's goal is to avoid being captured. The cop and the robber each know where the other is throughout the game. A graph is copwin if there is a strategy for the cop that guarantees capturing the robber. Many variants of the cop and robber game have been proposed [3].

The graph locating problem also dates from 1976, independently posed by Slater [9] and Harary and Melter [4]. It can be described as a robber game in which an immobile robber is hiding at an unknown vertex. The cop first chooses a set of vertices to be probes, then receives the distances from each of these probe to the robber simultaneously. From these distances the cop must locate the robber. The minimum number of probes needed to guarantee location is the metric dimension. A survey is included in [1], who note that there is also a sequential version of this problem. Here the cop chooses one probe at a time and receives the distance from this probe to the robber before having to choose the next probe. The sequential locating problem is studied further in [8].

In this paper we synthesize the two games into one, the robber locating game. We vary the cop and robber game so that, rather than knowing where the robber is, the cop has a sonar-like probe that gives the distance to the robber, and that can be activated from any vertex. Equivalently, we vary the sequential locating problem by allowing the robber to move to an adjacent vertex between successive probes.

2. The robber locating game

Assume throughout that \( G \) is a simple connected \( n \)-vertex graph, with \( n \geq 2 \). The robber locating game involves a cop, who can probe from anywhere, and a robber, who can only move to adjacent vertices. At the start, the robber chooses a
vertex $h_1$ to hide on. The cop chooses a probe vertex $p_1$ and receives the distance $d_1$ from $p_1$ to $h_1$. If the cop can locate the robber with this probe by uniquely determining where the robber is in $V(G)$, then the cop wins. Otherwise the robber may stay put or move to an adjacent vertex but may not move to the previous probe vertex; i.e., the robber’s second location $h_2$ satisfies $h_2 \in N[h_1] - p_1$, where $N[h_1]$ denotes the closed neighbourhood of $h_1$. Next the cop chooses a second probe, and the game continues as above. The cop wins if, after some probe $p_k$, she can locate the robber at $h_k$; the robber wins if this never occurs. A graph is locatable if there is a cop strategy that guarantees a win in a finite number of probes no matter what the robber does. For locatable graphs, the location number, $\text{loc}(G)$, is the minimum number of probes needed to guarantee locating the robber. As in the cop and robber game, we are interested in determining which graphs are locatable, and, for such graphs, finding the location number.

**Proposition 2.1.** A graph $G$ is locatable with $\text{loc}(G) = 1$ if and only if $G$ is a path.

**Proof.** $G$ is locatable with $\text{loc}(G) = 1$ if and only if for some vertex $p_1$, every vertex is a unique distance from $p_1$. This occurs if and only if $G$ is a path with $p_1$ as an endpoint. □

**Proposition 2.2.** $K_3$ and $K_{2,3}$ are locatable with $\text{loc}(K_3) = 2$ and $\text{loc}(K_{2,3}) = 3$.

**Proof.** Let the vertex set of $K_3$ be $\{a, b, c\}$. We give a cop strategy to locate the robber within two probes. Choose $a$ as the first probe. If $d_1 = 0$, then the robber is located at $a$; otherwise $d_1 = 1$, he is at $b$ or $c$, and he can move between $b$ and $c$ but not to $a$. Choose $b$ as the second probe. Now $d_2$ is 0 or 1, which locates the robber at $b$ or $c$, respectively.

Next, let the bipartition of $K_{2,3}$ be $(\{a, b\}, \{c, d, e\})$. We give a cop strategy to locate the robber within three probes; it is tedious but straightforward to check that no strategy can guarantee locating him with only two probes. Choose $c$ as the first probe. If $d_1 = 0$, then the robber is located at $c$. Otherwise, either $d_1 = 1$ and he is in $\{a, b\}$, or $d_1 = 2$ and he is in $\{d, e\}$. In either case, after his move he could be at any vertex in $\{a, b, d, e\}$ but not at $c$, so choose $a$ as the second probe. If $d_2$ is 0 or 2, then he is located at $a$ or $b$, respectively. Otherwise $d_1 = 1$ and he is in $\{d, e\}$, so after his move he could be at any vertex of $\{b, d, e\}$ but not in $\{a, c\}$. Choose $d$ as the third probe. Now $d_3$ is 0, 1, or 2, and he is located at $d, b,$ or $e$, respectively. □

**Proposition 2.3.** If $G$ contains $K_4$ as a subgraph, then $G$ is not locatable.

**Proof.** We give the robber a winning strategy that stays always on a fixed copy of $K_4$ in $G$. Let $p_k$ be the $k$th probe in $G$, and let $V_k = V(K_4) - p_{k-1}$, for $k \geq 1$. We prove by induction on $k$ that after the $k$th probe there exists some distance $d_k$ such that $\{x \in V_k : d(p_k, x) = d_k\}$ contains at least two vertices. It follows that for every cop strategy $S$ there exists a sequence of moves within $K_4$ for the robber that avoids location by $S$.

Suppose that $k \geq 1$, the robber is on $K_4$, and he has not been located in the first $k - 1$ probes. After his $(k - 1)$th move he could be at any vertex in $V_k$. Let $d = \min\{d(p_k, x) : x \in V_k\}$. Since $V_k$ contains at least three vertices of $K_4$, either $\{x \in V_k : d(p_k, x) = d\}$ or $\{x \in V_k : d(p_k, x) = d + 1\}$ contains at least two vertices; let $d_k = d$ or $d_k = d + 1$, respectively. □

**Proposition 2.4.** If $G$ contains $K_{3,3}$ as an induced subgraph, then $G$ is not locatable.

**Proof.** We give the robber a winning strategy that stays always on a fixed copy of $K_{3,3}$ in $G$. Let $(A, B)$ be the bipartition of $K_{3,3}$. Let $p_k$ be the $k$th probe in $G$, and let $V_k$ be the set of all possible locations for the robber on $K_{3,3}$ after his $(k - 1)$th move, for $k \geq 1$. We prove by induction on $k$ that for every choice of $p_k$ there exists some distance $d_k$ such that the set $X_k$ defined by $X_k = \{x \in V_k : d(p_k, x) = d_k\}$ contains at least two vertices, and if the robber is in $X_k$ at the $k$th probe, then the set $V_{k+1}$ of all vertices of $K_{3,3}$ where he could be after his next move contains at least four vertices. It follows that for every cop strategy $S$ there exists a sequence of moves within $K_{3,3}$ for the robber that avoids location by $S$.

Suppose that $k \geq 1$ and $V_k$ contains at least four vertices. Let $d = \min\{d(p_k, x) : x \in V_k\}$ and $X = \{x \in V_k : d(p_k, x) = d\}$. If $X$ contains at least two vertices of $V_k$, then let $d_k = d$. Otherwise $X = \{v\}$ for some vertex $v \in V_k$, and all three neighbours of $v$ in $K_{3,3}$ are at the same distance $d + 1$ from $p_k$. With the remaining two vertices of $K_{3,3}$ at distance $d + 1$ or $d + 2$. Since $V_k - v$ contains at least three vertices, either $\{x \in V_k : d(p_k, x) = d + 1\}$ or $\{x \in V_k : d(p_k, x) = d + 2\}$ contains two or more vertices; let $d_k = d + 1$ or $d_k = d + 2$, respectively. In each case $X_k$ contains at least two vertices of $K_{3,3}$, which means that at least five of the vertices of $K_{3,3}$ are in $X_k$ or are adjacent to a vertex of $X_k$. It follows that, if the robber can be at any vertex of $X_k$ at the $k$th probe, then after his $k$th move there are at least four vertices of $K_{3,3} - p_k$ where he can be. Thus $V_{k+1}$ contains at least four vertices. □

**Proposition 2.5.** $C_4$ is locatable with $\text{loc}(C_4) = 2$.

**Proof.** Let the vertices of $C_4$ be $a, b, c, d$ in order. Choose $a$ as the first probe. If $d_1$ is 0 or 2, then the robber is located at $a$ or $c$, respectively; otherwise he must be at $b$ or $d$ and cannot move to $a$. Choose $b$ as the second probe. Now $d_2$ must be 0, 1, or 2, which locates the robber at $b$, $c$, or $d$, respectively. □

**Proposition 2.6.** $C_5$ is not locatable.
Theorem 3.2. Let \( r \) be a tree with root \( r \). Let \( T \) be a tree with root \( r \). Let \( T \). The robber locating game for trees on \( T \) or \( T' \) is located on \( T \) or \( T' \). The bound in Theorem 3.2 is sharp for spiders.
Theorem 3.3. If $T$ is an $n$-vertex spider with $n \geq 3$, then $\text{loc}(T) = \Delta(T) - 1$.

Proof. Since $T$ is a spider, there exists a vertex $r$ with $\text{deg}(r) = \Delta(T) \geq 2$, which we choose as the root. Each other vertex is on exactly one of the $\Delta(T)$ threads at $r$. We define recursively a strategy $S_r$ with the RLP for $r$ that locates the robber in at most $\Delta(T) - 1$ probes. The result then follows from Theorem 3.2, Lemma 3.1, and induction on $n$.

Step 1. Choose a leaf $w$ at the end of a thread $L$ as the first probe. If $\Delta(T) = 2$, then $T$ is a path with $w$ an endpoint, so this probe locates the robber by Proposition 2.1. Hence we may assume $\Delta(T) > 2$. If the robber is on $L$, then stop: he is located in one probe. Hence we may assume the robber is not on $L$.

Step 2. Define $T'$ to be $T - (L - r)$ (i.e., remove the thread up to but not including $r$). Now $\Delta(T') = \Delta(T) - 1$. Apply strategy $S_r$ to $T'$ to locate the robber in at most $\Delta(T) - 2$ more probes, for a total of at most $\Delta(T) - 1$ probes. \hfill $\square$

Corollary 3.4. $\text{loc}(K_{n-1, 1}) = n - 2$ for $n \geq 3$.

Theorem 3.5. If $T$ is a tree with $n \geq 3$ vertices, then $\text{loc}(T) \leq n - 2$, with equality if and only if $T = K_{n-1, 1}$.

Proof. Choose a root $r$ for $T$. We define recursively a strategy $S$ with the RLP for $r$ that locates the robber in $n - 2$ probes when $T = K_{n-1, 1}$ and less than $n - 2$ probes otherwise. The result then follows from Theorem 3.2, Lemma 3.1, and induction on $n$.

For any subtree $T'$ of $T$, let $n(T')$ be the number of vertices in $T'$, and let $\text{pr}(T')$ be the maximum number of probes to locate the robber on $T'$ using strategy $S$. We assume $\text{pr}(T') \leq n(T') - 2$ for every proper subtree $T'$ of $T$ with $n(T') \geq 3$, and we want to show $\text{pr}(T) \leq n - 2$.

Step 1. Suppose that $T$ is a path. Choose either endpoint as $p_1$. By Proposition 2.1, $\text{pr}(T) = 1 \leq n - 2$, with equality if and only if $T = K_{2, 1}$.

Step 2. Suppose that $T$ is not a path and that there is a thread $L$ at some vertex $v$ with $\text{deg}(v) \geq 3$ such that $r$ is on $L$. Let $w$ be the leaf at the end of $L$. Choose $p_1 = w$. If the robber is on the thread $L$, then he is located in one probe. Otherwise he is not on $L$ (and thus not at $r$); in this case let $T' = T - (L - v)$. Now $n(T') \geq 3$, since $v$ has at least two neighbours in $T - L$. Apply strategy $S$ to $T'$ with root $v$ to get $\text{pr}(T') \leq 1 + \text{pr}(T') \leq n(T') - 1 \leq (n - 1) - 1 = n - 2$, with equality if only if $T' = K_{n-2, 1}$ and $w$ is adjacent to $v$; i.e., $T = K_{n-1, 1}$.

Step 3. We may now assume that $r$ is not on any thread in $T$; in particular, $\text{deg}(r) > 1$ and there are no threads at $r$, so $n \geq 7$. Choose $p_1 = r$, so the RLP holds for $r$. Let $d_r$ be the distance from $r$ to the robber. If $d_r = 0$, then the robber is located at $r$ in one probe, where $1 < n - 2$. Otherwise, since $p_1 = r$, he cannot now move to $r$, so the RLP will continue to hold for $r$ at the next probe.

For every vertex $v \neq r$ of $T$, let $T_v$ be the proper subtree of $T$ rooted at $v$ that is the component of $T - e$ containing $v$, where $e$ is the first edge on the path from $v$ to $r$ in $T$. Let $s$ be a neighbour of $r$ such that $n(T_v)$ is minimized and let $s'$ be any other neighbour of $r$, so $n(T_v) \leq n(T_{s'})$. Now $n(T - T_v) \geq n(T_v) + 1$ and $n = n(T_v) + n(T - T_v)$ so $n - 2 \geq 2n(T_v) - 1$. Let $k = \text{deg}(s) - 1$ and let $t_1, \ldots, t_k$ be the neighbours of $s$ other than $r$ in descending order of $n(T_v)$. If $n(T_v) = 1$, then go to Step 4; otherwise, let $i = 1$ and go to Step 5.

Step 4. Suppose $n(T_v) = 1$, so $n(T_v) = 1$ for $1 \leq j \leq k$ and $t_1, \ldots, t_k$ are all leaves. Since there are no threads at $r$, we have $k > 1$ and so $n(T_v) \geq 3$. Choose $p_2 = t_1$ and let $d_{t_1}$ be the distance from $t_1$ to the robber. If $d_{t_1} = 0$ or 1, then he is located at $t_1$ or $s$ in two probes, where $2 < n - 2$. If $d_{t_1} = 2$, then he is on $T_v - t_1$ but not at $s$, so apply strategy $S$ to $T_v - t_1$ with root $s$ to get $\text{pr}(T) \leq 2 + \text{pr}(T_v - t_1) \leq n(T_v) - 1 \leq n - 2$. Finally, if $d_{t_1} \geq 3$, then the robber must be on $T - T_v$ but not at $r$, so apply strategy $S$ to $T - T_v$ with root $r$ to get $\text{pr}(T) \leq 2 + \text{pr}(T_v - t_1) \leq n(T_v) - 1 \leq n - 2$.

Step 5. We may now assume the following:

- $1 \leq i \leq k$, $n(T_v) \geq 2$, and there have been $2i - 1$ probes so far;
- the last probe was $r$, and the distance from $r$ to the robber was $d_r \geq 1$;
- the robber is not on $T^*$, where $T^* = T_1 \cup \cdots \cup T_{i-1}$;
- if $i > 1$, then the robber is not at $s$.

Since $n(T_v) \geq 2$ for $1 \leq j \leq i$, we have $n(T^*) \geq 2i - 2$ and $n(T_v) \geq 2i + 1$. Choose $p_{2i} = t_i$ and let $d_{t_i}$ be the distance from $t_i$ to the robber.

Step 5a. If $d_{t_i} = 0$, then the robber is located at $t_i$, with $\text{pr}(T) \leq 2i < n(T_v) < n - 2$.

Step 5b. If $d_{t_i} = 2$, then the robber must be on $T_v - T^*$ but not at $s$, so apply strategy $S$ to $T_v - T^*$ with root $s$ to get $\text{pr}(T) \leq 2i + \text{pr}(T_v - T^*) < n(T^*) + 2 + n(T_v - T^*) - 2 = n(T_v) < n - 2$.

Step 5c. If $d_{t_i} = 1$, then the robber is at a neighbour of $t_i$, which may or may not be $s$, so after his move he will be either at $r$ or on $T_v$, but not at $t_i$. Choose $p_{2i+1} = r$ and let $d_{t_i}'$ be the current distance from $r$ to the robber.

Case (i) If $d_{t_i}' = 0$ or 1, then he is located at $r$ or $s$, respectively, and $\text{pr}(T) \leq 2i + 1 \leq n(T_v) < n - 2$.

Case (ii) If $d_{t_i}' = 2$ or 1, then he is on $T_v - T_i$ but not at $s$, so apply strategy $S$ to $T_v - T_i$ with root $s$ to get $\text{pr}(T) \leq 2i + 1 + \text{pr}(T_v - T_i) \leq n(T_v) + n(T_v - T_i) - 2 < n - 2$.
of degree 2, does it follow that either $G$ is not locatable or $T$ contains a vertex of degree 2 and replacing it with an edge between its two neighbors? Is this true in the particular case where $G$ is a tree? The strategy in Section 3 for trees may use more probes than necessary. The problem of finding a strategy for trees that uses $\text{loc}(T)$ probes for each tree $T$ remains, as does finding a better bound for the location number of trees in terms of parameters other than $n$.

This paper begins the study of the robber locating game, but there are many more questions to explore. There are also variants of this game; for example, Erickson et al. [2] have obtained some results for a relaxed version in which the robber is permitted to move to the probe vertex.

Acknowledgements

Thank you to members of the audiences at the CanaDAM and ECCC conferences, to the referees, and to the editor for their helpful comments and suggestions.

References