On the Capture Time of Cops and Robbers Game on a Planar Graph

Photchchara Pisantechakool(✉) and Xuehou Tan

School of Science and Technology, School of Information Science and Technology, Tokai University, 4-1-1 Kitakaname, Hiratsuka 259-1292, Japan
3btad008@mail.tokai-u.jp

Abstract. This paper examines the capture time of a planar graph in a pursuit-evasion games' variant called the cops and robbers game. Since any planar graph requires at most three cops, we study the capture time of a planar graph G of n vertices using three cops, which is denoted by $capt_3(G)$. We present a new capture strategy and show that $capt_3(G) \leq 2n$. This is the first result on $capt_3(G)$.

Keywords: Pursuit-evasion game · Cops and robber game · Capture time · Game length

1 Introduction

Pursuit-evasion games are turn-based ones in which one player, controlling the "evader", tries to avoid being captured by the "pursuers" controlled by another player. In each round, two players take turns to move their pieces. The games have many versions, varied through many means; such as environments, knowledge of terrain and the opponent's locations, or movements. The pursuers win if they can capture the evader, and the evader wins if he can avoid being captured indefinitely.

There are two well-known variants of pursuit-evasion games in a graph setting. In a variant where the pursuers do not know the location of the evader (see Gal [6], and Parsons [12,13]), the problem is similar to "graph searching problem". The difference is that in the pursuit-evasion game, the evader may move from an unexplored area to some areas that had been visited or cleared by the pursuers. To prevent such recontaminations, some pursuers are required to stand guard while others are searching. Finding the number of pursuers to successfully clear the graph has been the main focus of the problem. There are two approaches in finding the number of pursuers, based on searching strategy; non-monotonic search, which allows recontaminations (Kehagias et al.'s [7]), and monotonic search, which does not (such as Bienstock's [4], LaPaugh's [8] and many in Alspach's survey [2]).

This work was partially supported by JSPS KAKENHI Grant Number 15K00023.

T-H.H. Chan et al. (Eds.): COCOA 2016, LNCS 10043, pp. 3–17, 2016.
DOI: 10.1007/978-3-319-48749-6_1

Another variant is often called a "cops and robbers game". In this variant, both players have full knowledge of the terrain and the opponent's locations. The earlier researches focused on *cop number* or how many cops are needed to win the game in a given setting (such as Nowakowski and Winkler's [11] and Quilliot's [14]). In a graph setting, Aigner and Fromme [1] proved that some instances of this game can be won by the cop player controlling one cop; such instances are called cop-win graphs. Also, they proved that for a planar graph, three cops always suffice to win. So, the cop number for a planar graph is at most three. Maurer et al. provided a report about the cops and robbers game on many different planar graphs [9].

The length of the game, or the capture time of a graph by j cops, denoted as $capt_j(G)$, has been studied recently. In 2009, Bonato et al. [5] considered the capture time of various cop-win graphs, and concluded that while the capture time of a cop-win graph of n vertices is bounded above by $n-3$, half the number of vertices is sufficient for a large class of graphs including chordal graphs. For the graphs with multiple cops required to win, the capture time can be calculated by a polynomial-time algorithm if the number of cops is fixed. They also proved that the problem of determining the minimum number of cops needed to win under the time constraint is NP-complete. In 2011, Mehrabian [10] showed that the capture time of grids, whose cop number is known to be two, is half the diameter of the graph, or $capt_2(G) \leq \lfloor \frac{m+n}{2} \rfloor - 1$ for $m \times n$ grid.

Aigner and Fromme [1] have proved that any planar graph requires at most three cops. In their proof, they introduced two concepts which can be used in actual capture strategy. One concept is the assignment of a stage i in the capture strategy to a certain subgraph $R_i \subset G$, which contains all the vertices that the robber can safely enter. The graph R_i is called the *robber territory*. Another is a concept called the *guarded path*, which is the shortest path between two vertices such that a cop can capture the robber if the robber ever enters any vertex on the path. After one cop successfully controlled a path, the robber territory changes such that $R_{i+1} \subsetneq R_i$. Their method is to repeatedly find a new guarded path that differs from previous ones, until the robber territory is eventually reduced to one vertex. However, the capture time of their strategy is not examined in [1].

In this paper, we focus on the capture time of the cops and robbers game on a planar graph in general, which has not yet been studied well. We present a new capture strategy by refining the work of Aigner and Fromme in the following two sides: (i) a new guarded path introduced at a stage shares only its end vertices with any current path, and (ii) the end vertices of a newly introduced guarded path are on or very close to some *outer cycle*, whose all vertices belong to the infinite face of the robber territory. These two refinements are involved, specially the second needs some deep observations. All guarded paths in our strategy are chosen so that they are almost distinct, excluding their end vertices and a special situation in which two new paths are simultaneously introduced at a stage. A strategy with capture time less than $2n$ can then be obtained. The capture time using our strategy is provably faster than $2n$.

In Sect. 2 of this paper, we introduce the definitions and notations for the Cops and Robbers game. In Sect. 3 we review the known results from Aigner and Fromme's research. We introduce the new concepts for our strategy in Sect. 4. Section 5 describes how to choose the guarded paths, so as to make our refinitions. The correctness and completeness of the capture strategy are provided in Sect. 6, and the capture time of our strategy is analyzed in Sect. 7. The conclusion is given in Sect. 8.

2 Preliminaries

The game of Cops and Robbers in this research is played on a planar graph, and both players know the locations of one another's pieces [1]. The cop player controls three cops, and the robber player controls one robber. The game starts by letting the cop player choose the vertices to occupy, or place her cops first, and then the robber player occupies his vertex. Such a cop-robber turn is often called a *Round*. The first cop-robber turn is considered as Round 0. For simplicity, we assume that all cops occupy the same vertex in Round 0. In one's turn or a round, a player can move his/her piece to only an adjacent vertex of that piece, and he/she can choose not to move as well. In her turn, the cop player can move up to three cops at the same time.

Let $capt_3(G)$ denote the capture time of the cops and robber game on a planar graph G. Clearly, if there is a capture strategy on G such that after a finite number k (>0) of rounds, the robber is captured (i.e., the vertex occupied by the robber is also occupied by a cop), then $capt_3(G) \leq k$.

In order to capture the robber, the cop player may employ a cop to prevent the robber from crossing a certain line, like a goalkeeper preventing a ball from entering the goal in soccer. Once she moved to the right spot on that line, she is occupied with moving along the line in reaction to the robber's movement. This kind of actions limits the robber's movement to one side of the line, and thus diminishes the area the robber can safely enter. For this purpose, we define below *stages* and *robber territories* as well.

Definition 1. *The stage i, $0 \leq i \leq t$, is the assignment of a subgraph R_i which has all the vertices the robber can still safely enter. The assignment of R_i is done after the cop player has fixed her pieces at the end of stage $i-1$, and we assume $R_0 = G$. The subgraph R_i is called the robber territory.*

We assume that stage 0 exactly coincides with Round 0, and thus $R_1 = G - \{e\}$ where e denotes the vertex initially occupied by the cops. At a stage i (>0), the cop player constructs one or two new guarded paths so that R_i is reduced to $R_{i+1} \subsetneq R_i$. Thus, stage i may consist of several rounds. The *length* of stage i is then defined as the number of rounds it takes for the cop player to fix her pieces, and the length of stage 0 is assumed to be zero. If we have a strategy that ends after t stage, the capture time of the strategy is equal to $\sum_{i=1}^{t}$ *length of stage* i. We will focus on the movements of cops because the length of a stage

is actually decided by the cop whose number of movements is the largest among three cops.

A graph $G = (V, E)$ is defined as a set $V(G)$ of vertices which are connected by a set $E(G)$ of edges. We assume the graph is planar and undirected. A *cut vertex* $v \in G$ is a vertex such that when removed, the graph G has the increased number of connected components. By the nature of our problem, graph G must be connected; otherwise the cops may not be on the same connected component as the robber and thus it may not be cop-win.

Denoted by $\pi(u, v)$ a shortest path between $u \in V(G)$ and $v \in V(G)$, and $|\pi(u, v)|$ the length of $\pi(u, v)$, measured by the number of edges in $\pi(u, v)$. The distance between two vertices u and v is then defined as $|\pi(u, v)|$. A cycle, denoted by C, is a path whose start vertex and end vertex are the same. The *diameter* of a graph G is the greatest distance among all pairs of vertices in G.

For a vertex $v \in V(G)$, $N(v)$ is defined as the set of the vertices adjacent to v. For a vertex set $S \subseteq V(G)$, $G[S]$ is defined as the subgraph of G that is induced by the vertex set S, i.e., an edge $e \in E(G)$ belongs to $G[S]$ if two vertices of e belong to S.

3 Known Results

In this section, we review some known results from Aigner and Fromme's work [1].

Lemma 1 *(1-cop win [Aigner and Fromme]). Graph G is a 1-cop-win if and only if by successively removing pitfalls, G can be reduced to a single vertex.*

A pitfall is a vertex $v \in V(G)$ such that there exists a vertex $u \in N(v)$ and $N(v) \subset N(u)$. Lemma 1 can be applied to any tree, since we can simply chase down the robber from some vertex to a dead end at some leaf.

Lemma 2 *(Planar Graph Cop Number [Aigner and Fromme]). For any planar graph G, the cop number of G is at most three.*

Their proof of Lemma 2 provided a series of concepts and tools that can be used in the actual strategy. One is the concept of the robber territory, which we introduced in Definition 2. The other concept, of *guarded paths*, is devoted to diminishing the area of the robber territory.

Lemma 3 *(Guarded Shortest Path [Aigner and Fromme]). Let G be any graph, $u, v \in V(G)$, $u \neq v$ and $P = \pi(u, v)$. We assume that at least two cops are in the play. Then a single cop c on P can, after the movements no more than twice the diameter of G, prevents the robber r from entering P. That is, r will immediately be caught if he moves into P.*

It is imperative that we provide the proof of Lemma 3 as well, particularly for the new claim that "a single cop c on P can, after the movements *no more than twice the diameter of G*, prevents r from entering P".

Modified Proof of Lemma 3. Suppose the cop c is on vertex $i \in P$ and the robber r is on vertex $j \in V(G)$. Assume $\forall z \in P, |\pi(z,j)| \geq |\pi(z,i)|$; denote this as (*).

Claim A. No matter what the robber does, the cop, by moving in the appropriate direction on P, can preserve condition (*). If the r does not move, then neither does c, and (*) holds. If r moves to a new vertex k, then $\forall z \in P, |\pi(k,z)| \geq |\pi(j,z)| - 1 \geq |\pi(i,z)| - 1$. If $i' \in P$ exists with $|\pi(k,i')| \geq |\pi(i,i')| - 1$, then c, by moving on P toward i', also reduces the distance by 1 and (*) still holds. Suppose there exist vertices $x, y \in P$ such that they are on the different sides of i on the path P, and $|\pi(k,x)| = |\pi(i,x)-1|$, $|\pi(k,y)| \leq |\pi(i,y)|$ or $|\pi(k,x)| \leq |\pi(i,x)|$, $|\pi(k,y)| = |\pi(i,y)| - 1$. This is impossible, since by the triangle inequality and minimality of P;
$|\pi(x,y)| \leq |\pi(k,x)| + |\pi(k,y)| \leq |\pi(i,x)| + |\pi(i,y)| - 1 = |\pi(x,y)| - 1$; a contradiction.

Claim B. It takes a number of movements no more than twice the diameter of G for c to enforce (*). First, c moves to some $i \in P$, which takes at most the diameter of G. By the same argument as described above, $|\pi(j,z)| < |\pi(i,z)|$ only holds for z's on P on one side of i. By moving in the direction of z which takes at most $|P|$ or the diameter of G moves, (*) is eventually forced. □

At a single stage, an unoccupied (free) cop c will move to position herself on some vertex of a guarded path P so as to enforce the condition (*), i.e., be on the vertex such that she takes less time than the robber to move to any other vertex on P. Until (*) is enforced by c, the robber r may cross P safely. Once (*) is enforced and constantly preserved by c, r can no longer cross P without being captured. The location satisfying (*) on P changes as the robber moves, but there always exists at least one at a given time.

Corollary 1. *If a cop c is already on P, then the number of rounds it take for c to eventually enforce (*) is bounded above by the length of P.*

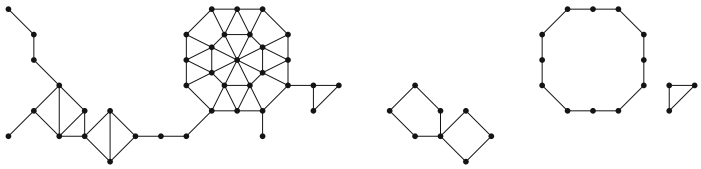
Corollary 2. *When a cop successfully controls (i.e., enforce (*) on) a shortest path, all vertices of that path do not belong to the robber territory.*

From Lemma 3, capturing the robber can be done by letting the cops alternatively take the role of a free cop and guard a new shortest path within the robber territory. Once the free cop successfully guards the path, she becomes occupied and another cop, whose guarded path no longer interacts with the robber territory, becomes the free cop at the next stage. At stage i, the path guarded at stage $i-1$ by a now-free cop is called an *obsolete path.*

Remark. From Lemma 3, the length of each stage of Aigner and Fromme's strategy is bounded by twice the diameter of R_i, or loosely by $2|V(R_i)|$, as their guarded paths are usually not distinct [1]. Suppose each stage only reduces one vertex in the worst case. Then, the capture time of Aigner and Fromme's strategy can roughly be bounded by $\sum_{i=1}^{n} 2i = n(n-1)$. Their capture time may actually be much faster than $O(n^2)$, but it needs a careful calculation (such as more detailed evaluation on how paths with shared vertices interact with the movements of the cops at each stage).

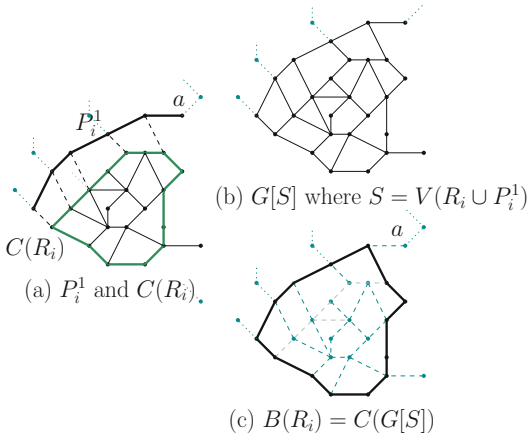
4 New Concepts

In order to establish a better upper bound on $capt_3(G)$, we make two refinitions over [1]. The first one is that a new path shares only its end vertices with any current path. It is worth pointing out that in our strategy, two new paths may also be introduced at a stage. As we will see, the guarded paths in our strategy are almost distinct, excluding their starting/ending vertices. (In [1], a new path may share more than just end vertices with a current path.). Our second refinition is to choose the end vertices of the new guarded path to be on or very close to the infinite face of robber territory. To precisely define where to choose the end vertices, we need a new concept called *outer cycles*.



(a): Robber Territory R_i (b): Outer cycles $C(R_i)$

Fig. 1. A robber territory (a) and its outer cycles (b).



(a) P_i^1 and $C(R_i)$ (b) $G[S]$ where $S = V(R_i \cup P_i^1)$

(c) $B(R_i) = C(G[S])$

Fig. 2. In (a) the current guarded path P_i^1 is shown in thick black line, and the outer cycle of R_i ($C(R_i)$) is shown in thick shaded cycle. (b) shows the subgraph induced on G by the vertex set $V(R_i \cup P_i^1)$, drawn in black lines and dots. The thick cycle in (c) represents the graph $B(R_i) = C(G[V(R_i \cup P_i^1)])$.

Definition 2. *The subgraph $C(R_i)$ of R_i is defined as the set of the outer cycles, whose all the vertices and edges belong to the* infinite *(exterior) face of R_i (Fig. 1). In the case of a polyhedral graph, where all of its faces can be considered as interior ones, we can choose any face as the infinite face. For graphs with multiple planar embeddings, outer cycles are made from the infinite face of the planar straight line drawing.*

Suppose that at the beginning of stage i, one or two current paths are guarded by the cops so as to prevent the robber from leaving R_i. These paths will be denoted by P_i^1 and P_i^2. Since P_i^1 and P_i^2 are assigned at the end of R_{i-1}, $P_i^1 \cap R_i = \emptyset$ and $P_i^2 \cap R_i = \emptyset$. We assume that P_i^1 always exists, i.e., R_i $(i > 0)$ can NEVER be assigned without P_i^1. The newly introduced path(s) at stage i will be denoted by P or/and Q.

For two end vertices of a new guarded path, one may consider to choose them from $C(R_i)$. But, it is difficult or even impossible in some cases for the new path to share a common vertex with P_i^1 or P_i^2. To overcome this difficulty, we will select the end vertices from the outer cycle of the subgraph induced by the vertices of the union of R_i, P_i^1 and P_i^2.

Definition 3. *Let $S = V(R_i \cup P_i^1 \cup P_i^2)$. The graph $B(R_i)$ (of enlarged outer cycles) is defined as $C(G[S])$.*

Note that $B(R_i)$ consists of only cycles, and thus not all vertices of R_i, P_i^1 and P_i^2 belong to $B(R_i)$. For instance, the vertex a of P_i^1 (Fig. 2(a)) does not belong to $B(R_i)$. See Fig. 2(c).

5 Capture Strategy

This section focuses on how to choose the guarded paths at each stage of our strategy. The analysis on the capture time of our strategy will be given in Sect. 7. We first give two propositions which we use throughout our capture strategy.

Proposition 1: At the end of each stage i, we have at least one free cop.

Proposition 2: During our strategy, any guarded path introduced at stage i shares only its end vertices with each of the current paths. For the two guarded paths introduced at the same stage, they have a common end vertex, and may share a subpath starting from that common vertex.

Before we begin, keep in mind that at some stage the robber territory may become a tree; in this case, only one cop suffices (Lemma 1). The main idea of our strategy is to let at most two cops guard two different paths, and then employ the free cop(s) to guard the new path(s), which makes one of the current guarded paths obsolete and thus reduces the robber territory.

Our capture strategy consists of two phases.

1. Initial Phase: We first find a location to place the cops, based on the structure of the graph G. This phase also establishes the very first pair of the guarded paths, and thus goes from the start to the end of stage 1. When it is over, R_1 is reduced to R_2.

2. Recursive Phase: At a stage i ($\geq$2), we construct new guarded path(s) using a case-analysis method. At the end of stage i, R_i is reduced to R_{i+1}. We do this recursively until R_i becomes the tree case.

Before describing the initial and recursive phases, we first give a procedure, called *TwoCopsStart*, which is used when we have two free cops at stage i (hence P_i^2 does not exist). The procedure takes two inputs: a cycle C of $B(R_i)$ and a starting vertex $u \in C$. The procedure *TwoCopsStart* finds the other end vertex for either of the two new paths. It is done by first setting the pointers x and y such that $|\pi(u,x)|$ and $|\pi(u,y)|$ on C are at $\lfloor \frac{|C|}{3} \rfloor$. Recall that a portion of P_i^1 may be included in $C \subseteq B(R_i)$, but $P_i^1 \cap R_i = \emptyset$. Thus $P_i^1 \cap C$ is the only portion of C that is not in R_i. Since P_i^1 as well as $P_i^1 \cap C$ are shortest paths in R_{i-1}, the length of the path $C - (P_i^1 \cap C)$, which is in R_i, is at least $|P_i^1 \cap C|$. Hence, at least a half of C is in R_i, and either x or y is initially in R_i. *TwoCopsStart* then repeatedly checks whether the pointer x (or y) belongs to R_i; if *not*, x (y) is moved on C further away from u, until x (y) belongs to R_i. Finally, x and y are returned as v_1 and v_2, respectively. The outputs v_1 and v_2 will be used to construct the new paths $\pi(u,v_1)$ and $\pi(u,v_2)$ in the graph $G[V(R_i)+u]$.

Procedure: *TwoCopsStart*
Input: a cycle $C \subseteq B(R_i)$ and a vertex $u \in C$.
Output: two vertices v_1 and v_2 for the new guarded paths $\pi(u,v_1)$ and $\pi(u,v_2)$.

1. Set a pointer x at u, move x clockwise along C for $\lceil \frac{|C|}{3} \rceil$ vertices.
2. **While** x is not in R_i, **do** move x to the next vertex on C clockwise.
3. Set a pointer y at u, move y counterclockwise along C for $\lceil \frac{|C|}{3} \rceil$ vertices.
4. **While** y is not in R_i, **do** move y to the next vertex on C counterclockwise.
5. **Return** $v_1 \leftarrow x$ and $v_2 \leftarrow y$.

5.1 Initial Phase

The initial phase has the following two objectives: (i) find a vertex to place the cops at stage 0, and (ii) establish the first pair of the guarded paths at stage 1 or in R_1.

At stage 0, if $B(R_0)$ is empty, then we know that the graph is a tree, which can be easily dealt with as stated in Lemma 1. In the case that $B(R_0)$ is not empty, we choose a vertex $e_0 \in B(R_0)$ such that e_0 is not a cut vertex (for the simplicity of assigning R_1). We place all cops c_1, c_2 and c_3 at e_0, and then wait for the robber player to place his piece r on the graph.

Suppose r is now located at some vertex in $R_0 - e_0$. At stage 1, we have $R_1 = R_0 - e_0$ and $B(R_1) = B(R_0)$ (as P_1^1 is the vertex e_0 and $P_1^2 = \emptyset$). Note that $e_0 \in B(R_1)$ is on some cycle C_1 of $B(R_1)$. Using R_1, we execute *TwoCopsStart* by letting $C \leftarrow C_1$ and $u \leftarrow e_0$. After obtaining the outputs v_1 and v_2, we find the shortest paths $P = \pi(e_0, v_1)$ and $Q = \pi(e_0, v_2)$ in $G[V(R_1) + e_0]$ and send c_1 and c_2 to guard P and Q, respectively. Note that the length of stage 1 is mainly determined by the operation of moving a cop to a vertex on the shortest path and then enforcing condition (*) on that path, which can be done without concerning the movements of r.

At the end of the initial phase, if P and Q separate R_1 into two or more components, R_2 is then the connected component containing the robber r. Otherwise, $R_2 = G[V(R_1 - P - Q)]$. The reduced robber territory R_2 is either a subgraph with at least one outer cycle, or a tree. If it is the tree case, the robber can simply be captured (Lemma 1). Otherwise, we enter Recursive Phase.

5.2 Recursive Phase

In this phase, we recursively reduce the robber territory R_i into R_{i+1}, until R_i is a tree. The reduction of R_i into R_{i+1} is done by constructing and controlling one or two new guarded paths (Corollary 2).

Recall that the current paths P_i^1 and P_i^2 were given at the end of stage $i-1$, and R_i can never be assigned without P_i^1. We distinguish the following situations.

Case (a): $B(R_i) \cap P_i^1$ *has at most one vertex.*

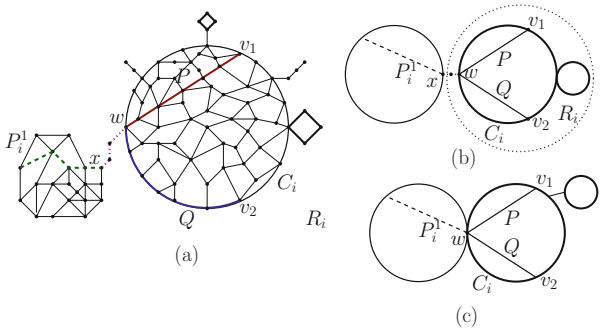


Fig. 3. An example of case (a); P_i^1, shown in thick dashed line, with the unique path $\pi(x, w)$, $x \in P_i^1$ and $w \in B(R_i)$. The new paths P and Q, with the common end vertex w, are shown in thick dark line.

Note that $P_i^2 = \emptyset$ in this case. If $B(R_i) \cap P_i^1 = \emptyset$, we find a vertex $x \in P_i^1$ and a vertex $w \in B(R_i)$ such that $|\pi(x, w)|$ is minimum among the shortest paths from a vertex of P_i^1 to the other of $B(R_i)$. Since $B(R_i) \cap P_i^1 = \emptyset$, the pair (x, w) is unique, and all vertices of $\pi(x, w)$ are cut vertices, see Fig. 3(b). We first move all three cops c_1, c_2 and c_3 to x, and then along $\pi(x, w)$ to u. In the case that $B(R_i) \cap P_i^1$ has one vertex, we let w be that vertex, see Fig. 3(c).

Let $C_i \subseteq B(R_i)$ be the cycle containing w. Since one cop can simply guard vertex w, two cops are free in this case. We execute *TwoCopsStart* procedure by letting $C \leftarrow C_i$ and $u \leftarrow w$. Note that w has some neighbors in R_i (two of them are on C_i). After obtaining the outputs v_1 and v_2 (which belong to R_i), we find the shortest paths $P = \pi(w, v_1)$ and $Q = \pi(w, v_2)$ in $G[V(R_i) + w]$, and then send the free cops, say, c_1 and c_2 to guard P and Q, respectively. In $G[V(R_i - P - Q)]$, the component containing r is then R_{i+1}.

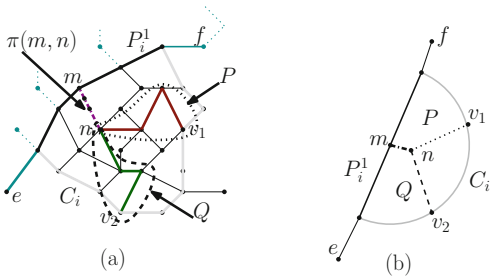


Fig. 4. An example of case (b); P_i^1 is shown in thick line, and $C_i \subseteq B(R_i)$ is shown in a combination of thick black line and thick gray lines. $P(m, v_1)$ is shown in thick dotted line, and $Q(m, v_2)$ in thick dashed line. P and Q may share some subpath $\pi(m, n)$, which is shown in thick dash dotted line.

Case (b): $B(R_i) \cap P_i^1$ *has more than one vertex and* $P_i^2 = \emptyset$.

Suppose $P_i^1 = \pi(e, f)$. Let $N(R_i)$ denote the set of all vertices u, where $u \in N(v)$ for some $v \in R_i$. We find $m \in N(R_i) \cap P_i^1$ such that $|\pi(m, f)| - |\pi(e, m)|$ (≥ 0) is minimal. Since $m \in P_i^1$, we also have $m \in B(R_i)$. Again, let $C_i \subseteq B(R_i)$ be the cycle containing m.

Since $B(R_i) \cap P_i^2 = \emptyset$, only one cop, say, c_1, needs to guard P_i^1. Thus two cops are free. We execute *TwoCopsStart* procedure by letting $C \leftarrow C_i$ and $u \leftarrow m$. After we obtain the outputs v_1 and v_2, we find the paths $P = \pi(m, v_1)$ and $Q = \pi(m, v_2)$ in $G[V(R_i) + m]$ and then send free cops c_2 and c_3 to guard P and Q, respectively. Note also that P and Q may share a common subpath if m has only one neighbor in R_i. See Fig. 4(a). In $G[V(R_i - P - Q)]$, the component containing r is then R_{i+1}.

Case (c): $B(R_i) \cap P_i^1$ *has more than one vertex and* $P_i^2 \neq \emptyset$.

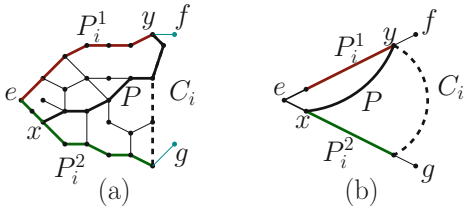


Fig. 5. An example of case (c); $P_i^1 = \pi(e, f)$, is shown in thick dark gray line, $P_i^2 = \pi(e, g)$ in thick dark line, and $P = \pi(x, y)$ in thick black line. $C_i \subseteq B(R_i)$ is shown in a combination of thick dashed lines and thick lines.

In this case, two cops have to guard P_i^1 and P_i^2, and thus we have only one free cop, say c_3. It can be deduced from Proposition 2 that P_i^1 and P_i^2 have a common vertex, say e. Let f (g) be the other end vertex of P_i^1 (P_i^2). By the denotations, $P_i^1 = \pi(e, f)$ and $P_i^2 = \pi(e, g)$. Note that f or g may not belong to $B(R_i)$ because f (g) may not be on any outer cycle of $G[V(R_i) \cup P_i^1 \cup P_i^2]$.

Next, we find two vertices $x \in N(R_i) \cap P_i^2$ and $y \in B(R_i) \cap P_i^1$ such that x and y are the vertices of P_i^2 and P_i^1, which are closest to e and f along P_i^2 and P_i^1, respectively. See Fig. 5 for example. Note that x has to be chosen from $N(R_i)$, instead of $B(R_i)$, because we want it to have some neighbor in R_i. It is also possible for $x = e$ in the case that $e \in N(R_i)$, and for $y = f$ when $f \in B(R_i)$. Finally, we find the shortest path $P = \pi(x, y)$ in $G[V(R_i) + x + y]$, and send the free cop c_3 to guard P. Again, R_{i+1} is the component containing r in $G[V(R_i - P)]$.

6 Correctness and Completeness

In this section we show that our capture strategy is correct and complete.

Theorem 1. *Proposition 1 is upheld for the whole of capture strategy.*

Proof. In case (a) and initial phase, two new paths are introduced and both start from a common vertex. Therefore, at least one cop is free at the next stage.

In case (b), as shown in Fig. 4(b), the new guarded paths P and Q may partition R_i into three components; each of them is a candidate of R_{i+1}, which is guarded by two cops at the next stage. Therefore, at least one cop is free at the next stage.

Similarly in case (c), as shown in Fig. 5(b), no matter which component becomes R_{i+1}, either P_i^1 or P_i^2 becomes obsolete and its cop is free at the next stage. □

Theorem 2. *At the end of each stage i of the robber territory, $R_{i+1} \subsetneq R_i$.*

Proof. It simply follows from the definition of the robber territory and Corollary 2. □

Theorem 3. *Proposition 2 is upheld for the whole of capture strategy.*

Proof. Using *TwoCopsStart* procedure, the two paths introduced in the initial phase, case (a) and case (b) always have a common vertex. For case (b), the newly introduced paths share exactly one vertex m with P_i^1, and two new paths may also share a subpath (e.g., $\pi(m, n)$ in Fig. 4).

In case (c), the newly introduced path $P = \pi(x, y)$ shares at most two common vertices x (when $x{=}e \in P_i^1$) and y with P_i^1, and exactly one common vertex x with P_i^2. □

The correctness of our strategy follows from Theorem 2, and the completeness follows from Theorems 1 and 3.

7 Capture Time of Our Strategy

In this section we describe in detail the movements of the cops during our strategy so as to get a better understanding of the capture time. We first introduce another concept, called the *active paths*.

Definition 4. *Active Path: Let $P = \pi(a, b)$ be a current guarded path at stage i, m and n the first vertex of P from a and b that has a neighbor in R_i, respectively. The subpath $P(m, n)$, from m to n, is called an active path of P.*

Sometimes a path P as whole may be an active path if both end vertices are in $N(R_i)$. If the path persists through many stages without becoming obsolete, the active portion may become smaller due to the change in robber territory. This active path can be represented as $B(R_j) \cap P$ $(j > i)$, since P is still a current path and thus the active portion of P at stage j belongs to $B(R_j)$.

Lemma 4. *Suppose path P is guarded by some cop c. Then it suffices for c to guard the active subpath of P.*

Proof. Let $P(m, n)$ be the active path of P in current stage i. By Definition 4, the robber territory R_i has no vertex adjacent to any vertices on $P - P(m, n)$. Suppose the robber r wants to travel to some vertex $u \in P - P(m, n)$. But u cannot be reached in R_i without traversing through P. The robber has to enter $P(m, n)$ first, which is a subpath of P, and by Lemma 3, he will get captured. □

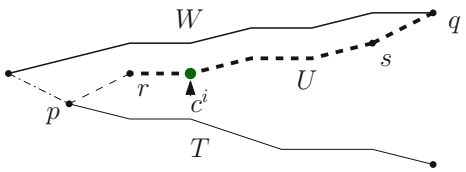
The guarding action of a cop requires three types of movements; (i) moving into the path, (ii) moving along the path to satisfy (*), and (iii) moving along the path while keeping to preserve (*). The length of a stage is mainly determined by the action of the free cop trying to control a new path. When a free cop successfully controls a path by enforcing (*) (i.e., be on the position that can move to any vertices on that path in smaller number of movements than the robber), the stage is then over. So (iii) movements can safely be ignored. In the following, we give a method to count the movements (i) and (ii) taken by the free cops.

In the initial phase, the cops are already on their own paths. The length of stage 1 involves only (ii) movements, which takes at most $max\{|P|, |Q|\}$. In the recursive phase, assume every stage $i > 1$ does not have any path that separates subgraph R_i into multiple components. This is the worst case because none of the vertices that do not belong to any guarded paths is removed from the robber territory.

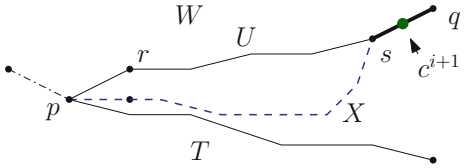
When a path, say, $U = \pi(p, q)$, is introduced at stage i (>1), it is traversed by a cop c^i at most $|U|$ (ii) movements (Lemma 3). See Fig. 6(a). As discussed above, the introduction of U makes one of the current paths, say W, obsolete at the end of stage i. At the next stage $i + 1$, a new free cop (differing from c^i), say, c^{i+1}, must move out of her obsolete path W (made obsolete by U) to a newly introduced path, say X. It might be faster to move c^{i+1} directly from W to X. But, for simplicity, we do the following: move to the common vertex of

W and U, and then (i.1) move along the intermediate path, which is a portion of the path (U), to reach destination path (X). See Fig. 6(b). At some later stage $i+j$ ($j>1$), U becomes obsolete by the introduction of some new path Y (at stage $i+j-1$) and the cop c^{i+j}, who was guarding U and labeled as c^i, moves to guard another new path Z. Note that Y can be X if U is obsolete at stage $i+2$. The cop c^{i+j} must first move out of the obsolete path (U) by (i.2) moving along the obsolete path to its vertex that is common with the current path (Y) at stage $i+j$, and then travels to Z using the method described above. See Fig. 6(c).

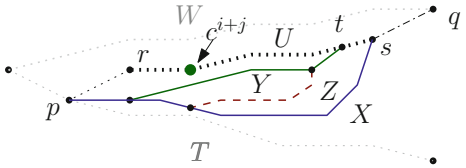
In summary, a path U is traversed by the free cops in three separate occasions: (ii) movements when being introduced at stage i, (i.1) movements when being a current guarded path at stage $i+1$, and (i.2) movements after becoming obsolete at stage $i+j$.



(a) Newly introduced U is traversed by c^i at stage i



(b) Current path U is traversed by c^{i+1} ($\neq c^i$) at stage $i+1$



(c) Obsolete path U is traversed by c^{i+j} ($=c^i$) at stage $i+j$

Fig. 6. A single path U is traversed by free cops on three separate occasions. At each stage, an obsolete path is shown in dotted line, a current path in normal heavy line, and a new path in dashed line. The portion of U traversed by a free cop is shown in thick line.

Let $U(r,s)$ be the active subpath of U. Following from Lemma 4, the cop c^i stops somewhere between r and s after she perform type (ii) movements. Thus, U is traversed by a free cop for at most $|U(p,s)|$ or $|U(r,q)|$ (ii) movements. In

order for c^{i+1} to move from obsolete path (W) to newly introduce path (X) at stage $i+1$, at most $|U(p,r)|$ or $|U(s,q)|$ (i.1) movements are made on U. At the beginning of stage $i+j$, U is made obsolete by Y, and the cop c^{i+j} must be somewhere between r and s on U. At stage $i+j-1$, the path Y is introduced, and the active portion of the current path U is $B(R_{i+j-1}) \cap U$. Thus, an end vertex of Y coincides with an end vertex, say, t of $B(R_{i+j-1}) \cap U$ (Proposition 2). From the discussion made above, t is somewhere between r and s on U. Therefore, U is traversed by the cop c^{i+j} for at most $|U(r,t)|$ or $|U(t,s)|$ (i.2) movements to move out of obsolete path U.

In conclusion, the number of the movements (i.1), (i.2) and (ii) on U is no more than $2|U|$. Hence, we have the following result.

Lemma 5. *In our capture strategy, each guarded path is traversed no more than twice of its length.*

Theorem 4. *In our capture strategy, all the paths used in evaluating the lengths of stages are distinct, excluding their end vertices.*

Proof. Supposed the guarded paths P_i and P_{i+1} are introduced during stage i and stage $i+1$, respectively. It follows from Theorem 3 that, excluding their end vertices, P_i is distinct from P_{i+1}. When two paths are introduced at the same stage and they may share a subpath (case (b)), only one of them (i.e., the one traversed by the free cop whose number of movements is larger) is used in evaluating the length of that stage. Therefore, the theorem follows. □

For completeness, in a tree case, the length of the chase on a tree is simply bounded above by the diameter of the tree.

Theorem 5. *For the cops and robbers game on a planar graph G of n vertices with three cops, $capt_3(G) \leq 2n$.*

Proof. The theorem directly follows from Theorem 4 and Lemma 5. □

8 Conclusion

We have presented a new capture strategy for the Cops and Robbers Game on a planar graph with three cops, and shown that the capture time of our strategy is no more than $2n$. This gives the first linear result on $capt_3(G)$.

An extension for future work is to apply our new concepts to improve a capture strategy in a polygonal environment; a problem researched by Bhadauria and Isler [3]. Their strategy, which is used to prove that three cops suffice in the polygonal environment with obstacles, is also based on the concepts given by Aigner and Fromme. The upper bound on the capture time is suggested to be as large as $2nA$, where A is the area and n is the number of vertices of the environment. In a geometric setting, even if all the paths are distinct, the sum of their lengths is still bounded above by A. It is rather loose as a large portion of the given environment is removed from the robber territory at the end of a stage. This thus suggests that by applying our strategy we may obtain a smaller upper bound, say, $2A$. We are working in this direction.

References

1. Aigner, M., Fromme, M.: A game of cops and robbers. Discrete Appl. Math. **8**(1), 1–12 (1984)
2. Alspach, B.: Searching and sweeping graphs: a brief survey. Le Mathematiche **59**, 2 (2004)
3. Bhaduaria, D., Klein, K., Isler, V., Suri, S.: Capturing an evader in polygonal environments with obstacles: the full visibility case. Int. J. Robot. Res. **31**(10), 1176–1189 (2012)
4. Bienstock, D., Seymour, P.: Monotonicity in graph searching. J. Algorithm **12**, 239–245 (2011)
5. Bonato, A., Golovach, P., Hahn, G., Kratochvíl, J.: The capture time of a graph. Discrete Math. **309**, 5588–5595 (2009)
6. Gal, I.: Search Games. Addison-Wesley, Reading (1982)
7. Kehagias, A., Hollinger, G.A., Singh, S.: A graph search algorithm for indoor pursuit-evasion. Math. Comput. Model. **50**(9–10), 1305–1317 (2008)
8. LaPaugh, A.S.: Recontamination does not help to search a graph. J. ACM **40**, 224–245 (1993)
9. Maurer, A., McCauley, J., Valeva, S.: Cops and robbers on planar graphs. In: Summer 2010 Interdisciplinary Research Experience for Undergraduates. University of Minnesota (2010)
10. Mehrabian, A.: The capture time of grids. Discrete Math. **311**, 102–105 (2011)
11. Nowakowski, R., Winkler, R.P.: Vertex-to-vertex pursuit in a graph. Discrete Math. **43**, 235–239 (1983)
12. Parsons, T.D.: Pursuit-evasion in a graph. In: Alavi, Y., Lick, D.R. (eds.) Theory and Applications of Graphs. Lecture Notes in Mathematics, vol. 642, pp. 426–441. Springer, Heidelberg (1978)
13. Parsons, T.D.: The search number of a connected graph. In: Proceedings of 9th South-Eastern Conference on Combinatorics Graph Theory and Computing, pp. 549–554 (1978)
14. Quilliot, A.: Some results about pursuit games on metric spaces obtained through graph theory techniques. Eur. J. Comb. **7**(1), 55–66 (1986)