The Role of Information in the Cop-Robber Game

Volkan Isler [a,*], Nikhil Karnad [b]

[a] *Department of Computer Science, Rensselaer Polytechnic Institute*
110 8^{th} Street, Troy, NY – 12180

[b] *Department of Computer Science, Rensselaer Polytechnic Institute*
110 8^{th} Street, Troy, NY – 12180

Abstract

We investigate the role of the information available to the players on the outcome of the cops and robbers game. This game takes place on a graph and players move along the edges in turns. The cops win the game if they can move onto the robber's vertex. In the standard formulation, it is assumed that the players can "see" each other at all times. A graph G is called cop-win if a single cop can capture the robber on G. We study the effect of reducing the cop's visibility. On the positive side, with a simple argument, we show that a cop with small or no visibility can capture the robber on any cop-win graph (even if the robber still has global visibility). On the negative side, we show that the reduction in cop's visibility can result in an exponential increase in the capture time. Finally, we start the investigation of the variant where the visibility powers of the two players are symmetric. We show that the cop can establish eye contact with the robber on any graph and present a sufficient condition for capture. In establishing this condition, we present a characterization of graphs on which a natural greedy pursuit strategy suffices for capturing the robber.

Key words: pursuit evasion games, limited visibility, greedy strategy

1 Introduction

A *Sensor Actuator Network (SAN)* is a network of devices equipped with sensing, communication, actuation (e.g. mobility) and computation capabilities.

* Corresponding author. Phone: +1-518-276-3275 Fax: +1-518-276-2529
Email addresses: `isler@cs.rpi.edu` (Volkan Isler), `karnan@cs.rpi.edu` (Nikhil Karnad).

SANs are becoming an inseparable component of automation systems. They are utilized in a wide range of automation tasks such as surveillance, quality inspection, search-and-rescue and environmental monitoring.

In a pursuit-evasion game, one or more pursuers try to capture an evader who, in turn, tries to avoid capture. Numerous SAN applications such as search-and-rescue, surveillance and tracking, can be modeled as pursuit-evasion games. For example, tracking, which is perhaps the primary use of a SAN, can be modeled as a pursuit-evasion game where the object being tracked is trying to evade by moving in a way that prevents the SAN from observing it.

There are two primary advantages in modeling SAN automation tasks as pursuit-evasion games. First, in most cases, the automation system works against an adversary. The game-theoretic formulation naturally captures the adversarial nature of the task. For example, surveillance robots must plan trajectories to detect burglars who will, in turn, develop strategies to avoid being detected. Second, pursuit-evasion strategies provide robust, worst-case guarantees in scenarios where the players are not necessarily adversarial. For example, a search-and-rescue team may utilize a pursuit strategy to find a lost hiker. In this case, even though the hiker is not adversarial, if a pursuit strategy exists, it guarantees that s/he will be rescued regardless of his or her actions. Such solutions are especially valuable when interacting with systems for which a good behavioral model is not available (e.g. humans).

Most pursuit-evasion games are well studied. However, solving games that model SAN applications requires addressing additional challenges which are not addressed in standard game formulations. Consequently, there has been recent growing interest in solving pursuit-evasion problems which address communication and sensing issues. For example, in [19] a pursuit-evasion game where a team of ground and aerial vehicles try to capture an evader is studied. The authors present a distributed systems architecture. Recently, Cao et al. studied an "asset protection game" where a sensor network team guards an area [8]. The authors address communications issues such as delay and packet losses.

In this paper, we investigate an aspect of pursuit-evasion games that is crucial in SAN applications: the information available to the pursuer. Most existing pursuit strategies require perfect information about the state of the game at all times. In other words, it is assumed that the pursuers have enough sensing capabilities to observe the evader's state at all times. However, since SANs are typically made up of inexpensive components which have limited sensing capabilities, solutions obtained for such models are not applicable to SAN applications.

In the present work, we investigate the role of the information available to the players on the outcome of the *cops and robbers game*. The environment in this game is represented by a graph. Players move along the edges in turns. The cops win the game if they can move on to the robber's vertex. In addition to obvious applications in surveillance and search-and-rescue, variants of the cops-and-robbers game have been used to model various network security problems [9]. Further, the study of the cops and robbers game sheds light onto graph theoretic properties such as vertex separation [10] and tree-width [5].

In this paper, we focus on the game where there is only a single cop. In the standard formulation, it is assumed that the players can see each other at all times. Here, we study the effect of reducing the pursuer's visibility.

Throughout the paper, we use the term "cop" interchangeably with "pursuer" and the term "robber" interchangeably with "evader".

2 Related Work

In the cops and robbers game, it is easy to see that if the robber has no visibility (that is, if it cannot gather information about the position of the cops), a single cop can capture the robber on any graph using a simple random-walk strategy. This gives an expected capture time of $O(nm^2)$ where n denotes the number of vertices and m denotes the number of edges. In [3], Aleliunas et al. presented a strategy for capturing the evader with additional restrictions on the space requirements for the pursuer strategy. Recently, Adler et al. revisited the game and showed that a single pursuer can catch the evader in $O(n \log n)$ time [1]. The strategy works even if the evader can jump from its current vertex to an arbitrary vertex. It was also shown that this analysis is tight: there are graphs and matching evader strategies which guarantee that no pursuer strategy can capture the evader in less than $\Omega(n \log n)$ steps.

The full visibility version (where the players know each others positions at all times) has received significant attention [17,18,7]. It is known that under the full-visibility model, the class of graphs on which a single pursuer suffices is the class of *dismantlable* graphs. The number of pursuers necessary to capture the evader on a graph G is known as the cop number of G. Aigner and Fromme showed that the cop number of planar graphs is at most 3 [2]. In [13], an analysis of the lengths of games on chordal graphs was presented. Further, in [14], Hahn et al provide an algorithmic characterization of reflexive cop-win digraphs when the cop-number k is fixed. The cop number of general graphs is open (c.f. [16,11,2]). The problem of determining whether k cops with given initial locations can capture a robber on a given undirected graph is EXPTIME-complete [12]. For the complexity of pursuit in directed graphs, see [12] and references therein.

Isler et al. studied the case where the evader has local visibility [15]. They study a variant where the players move simultaneously, and introduce the notion of i-visibility where a player with i-visibility can see another player only if the distance between them is at most i. It was shown that when the evader has 1-visibility (i.e. can see only the neighbors of its current location), two cops with 1-visibility can capture the evader with high probability on any graph. The expected capture time with two pursuers is polynomial in the number of vertices. A characterization of cop-win graphs where a single pursuer suffices to capture the evader was also presented. It was also shown that when the evader has 2-visibility, the number of cops required becomes unbounded: there are graphs which require $\tilde{\Omega}(\sqrt{n})$ cops to capture an evader with 2-visibility.

3 Our results

A graph G is called cop-win if a single cop can capture the robber on G. In this paper, we study the effect of reducing the cop's visibility on cop-win graphs. In particular, we study what happens if the cop can see the robber only if the distance between the two is less than or equal to some threshold value. Throughout the paper, we study connected, undirected graphs with self-loops (i.e. the players can stay in their vertices if they choose to). In the first part of the paper, we focus on the worst-case scenario, and assume that the robber has global visibility. Hence, it can see the cop at all times.

First, we obtain a positive result and show that a cop with small or no visibility can capture the evader on any cop-win graph (even if the robber still has global visibility). The cop uses a randomized strategy. We show that the capture probability within a finite time interval is non-zero. This yields an upper bound on the expected capture time which is exponential in the number of vertices (Section 5).

On the negative side, we obtain a lower bound and show that the increase in capture time is indeed exponential. We show that there exists a cop-win environment and a robber strategy such that the expected capture time for any cop strategy is exponential in the number of vertices (Section 6). More precisely, we show that for any n, there exists an environment with $O(n)$ vertices for which the expected capture time is $\Omega(3^n)$.

In the third part of the paper (Section 7), we initiate the study of the variant where the players' visibility ranges are the same [1]. First, we show that the pursuer can establish eye-contact with the evader on any graph. Then, we

[1] This version is suggested in [2].

study a natural greedy strategy and present an algorithmic characterization of graphs on which the greedy strategy suffices to capture the evader. These two results combined yield a sufficient condition for the cop to capture the robber in the symmetric visibility case.

4 Game model

We study the cops and robbers game on undirected graphs. Throughout the paper, we focus on the single cop version. The locations of the players are specified at the beginning of the game. The players move in turns. A move is considered as a transition from a vertex to any of its adjacent vertices. At each time step, first the evader moves along an edge. Next the pursuer moves. For a graph $G = (V, E)$, we use the following definition of neighborhood of a vertex $v \in V$:

$$N(v) = \{v\} \cup \{u|(v,u) \in E\}$$

Note that the neighborhood is closed, i.e. the vertex v is included in the neighborhood of itself. This means that the players can stay in their current vertices if they choose to. We use the notation $d(u, v)$ to denote the length of a shortest path (distance) between u and v. The cop captures the evader if he can move onto the evader's current vertex.

We say the pursuer has k-visibility, when the location of the evader is revealed to the pursuer only if the distance between them is at most k. A pursuer with global visibility can see the evader at all times. An evader with k-visibility is defined similarly.

5 Upper bound on capture time

Recall that a graph G is called cop-win if a global visibility pursuer can capture the evader on G. In this section, we show that a pursuer with no visibility can also capture the evader on a cop-win graph. The class of cop-win graphs admits a simple characterization:

Definition 1 (Dismantlable Graphs [18,7]). *Suppose i and j are nodes of a graph H such that $N(i) \subseteq N(j)$. The map that takes i to j and every other vertex of H onto itself is a homomorphism from H to $H - \{i\}$. This operation is called a fold of graph H and we say vertex i folded onto vertex j. A finite graph H is said to be dismantlable if there exists a sequence of folds reducing H to a graph with one vertex.*

A graph G is cop-win if and only if G is dismantlable [18,7].

Definition 2 (Folding Tree). *Let $G = (V, E)$ be a dismantlable graph. Given a vertex $v \in V$, a folding tree for G with respect to v is a tree rooted at v, which represents a folding sequence of G where v is the only remaining vertex. When a vertex i folds onto j in the sequence, j becomes the parent of i in the folding tree.*

We proceed as follows. First, we obtain an upper bound on the capture time for a pursuer with global visibility.

Lemma 3 *Let G be a dismantlable graph with n vertices. There exists a vertex v such that the cop with global visibility can start from v and capture the evader in at most n steps.*

In [15], it was shown that the pursuer can start at the root of a folding tree and chase the evader in such a way that (i) the pursuer remains an ancestor of the evader on the folding tree, and (ii) every time the evader revisits a leaf-to-root path P, the pursuer's height on P decreases. These two results combined imply that the pursuer never revisits a vertex during the game. This, in turn, implies Lemma 3.

Next, using this upper bound, we obtain a randomized (mixed) strategy for a pursuer with no visibility.

Theorem 4 *The expected time it takes for a pursuer with no visibility to capture an evader with global visibility on a dismantlable graph $G = (V, E)$ is at most $2n\Delta^n$, where Δ is the maximum degree of G, and $n = |V|$.*

PROOF. By Lemma 3, a pursuer with global visibility can start from the root of the folding tree and capture the evader in at most n steps. Which means that, no matter how the evader moves, there exists a sequence of pursuer moves, of length at most n, which guarantees capture. The pursuer with no visibility will travel to the root and then guess the sequence of evader moves. Since there are at most Δ possibilities at each step, the probability of a correct guess is at least $\frac{1}{\Delta^n}$. Hence, by repeating this process with independent guesses, the pursuer is expected to capture the evader in Δ^n trials. Each trial contains a trip to the root followed by a sequence of n moves. Since the length of each trial is at most $2n$, the expected capture time is at most $2n\Delta^n$. □

6 A lower bound on the capture time

Theorem 4 suggests the possibility of an exponential increase in expected capture time due to loss of visibility. In this section, we show that there exists a class of graphs and evader strategies on these graphs such that the expected capture time for any pursuer strategy is lower bounded by a quantity that is indeed exponential in the number of vertices.

We start with the case where the pursuer has no visibility. Later, we will extend the result to the case where the pursuer has k-visibility for arbitrary k. First we introduce the environment for which we will present an evader strategy.

6.1 Definitions

Given $n \in \mathbb{N}$, we construct a graph G_n as follows (Figure 1).

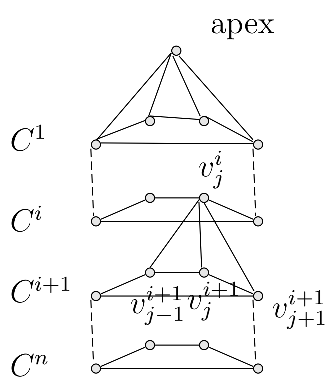

Fig. 1. Construction of the graph G_n, the environment on which our pursuit-evasion game is played.

Start with a cycle of length 4. Make n copies $C^1, C^2, \ldots, C^n$ and "stack" them one below the other where C^i is placed above C^{i+1}. We align the cycles so that for each vertex $v \in C^i$ there is a unique vertex in C^{i+1} which is directly below v. We can now rename the vertices in $C^i = \{v_1^i, \ldots, v_4^i\}$ such that v_j^i is *above* v_j^{i+1}. For each v_j^i, $i = 1, \ldots, n-1$, $j = 1, \ldots, 4$, we introduce edges to $v_{j-1}^{i+1}, v_j^{i+1}, v_{j+1}^{i+1}$ (We define $v_5^{i+1} = v_1^{i+1}$ and $v_0^{i+1} = v_4^{i+1}$.).

Finally, we add a special vertex to this graph and connect it to all vertices in C_1. We call this vertex the *apex* of our graph.

Definition 5 The environment *refers to G_n for a given n.*

Given two cycles C^i and C^j, we say that C^i is above (below) C^j if $i < j$ (respectively $i > j$). Similarly, we say v_k^i is above v_k^j if $i < j$. We say v_k^i is directly above v_k^j if $i = j - 1$. The players are *vertically aligned* if one player is on a vertex that is above the other player's vertex.

It should be noted that the environment is dismantlable and hence cop-win.

A *configuration* describes the state of the game by specifying the vertices that the pursuer and evader occupy.

Definition 6 **A configuration** *is given by the pair* (p, e) *where* p *is pursuer's current vertex and* e *is evader's current vertex.*

Recall that the evader is captured if at any point in time, the pursuer and the evader both occupy the same vertex of the graph.

We define $T[p, e]$, the *capture time* for a given configuration (p, e), as the number of moves required for the pursuer located at p to capture the evader located at e. Without loss of generality, we assume that the pursuer minimizes the capture time whereas the evader maximizes it. The capture time for a graph $G = (V, E)$ is given by $\max_{p,e \in V} T[p, e]$. In what follows, we obtain a lower bound on the capture time of G_n. First, let us take a closer look at the global visibility game and show the unique pursuer strategy to capture the evader in this case.

6.2 Capture on G_n with a global visibility pursuer

In this section, we show how an optimal strategy for a pursuer with global visibility can capture an evader with global visibility on G_n. The intuitive idea for this pursuer strategy stems from the fact that if the players are not vertically aligned at any point in the game, then the evader resets the progress made by the pursuer thus far by escaping to the top level.

Consider a configuration (p, e). There are three possibilities.

Case (i) The pursuer is at or below the evader's level. The evader avoids capture by staying on his level while moving away from the pursuer horizontally.

Case (ii.a) The pursuer is above the evader and the players are *vertically aligned*. Irrespective of the evader move, the pursuer can move down and maintain vertical alignment.

Case (ii.b) The pursuer is above the evader and the players are not vertically aligned. The evader can *reset* the game by exploiting the horizontal separation between the players and moving up diagonally. The only way for the pursuer to establish horizontal alignment is to move up to the apex.

Thus, an optimal pursuit strategy would avoid configurations in Case (i) and (ii.b) until capture. This yields the following theorem.

Theorem 7 *In an optimal pursuer strategy, the pursuer remains above the evader until capture. Moreover, there exists an optimal evader strategy where the capture occurs on C^n.*

In the next section, we obtain a lower bound on the expected capture time with a pursuer with zero-visibility.

6.3 Evader Strategy

In this section, we describe an evader strategy S against a pursuer with no visibility. We will then show that the expected capture time of any pursuer strategy on G_n against the S is large.

Recall that to capture the evader, the pursuer must remain above it (Theorem 7). Since the pursuer with zero-visibility has no way of knowing where the evader is at the end of each round, he simply has to make a guess to maintain this invariant. The evader strategy S then penalizes each wrong guess by resetting the game.

The evader strategy S randomizes how the evader moves down the stack by picking, with equal probability, one of three neighboring vertices at the lower level. To capture the evader, the pursuer will have to make n *consecutive* "correct" guesses.

The crucial component of S is to penalize a single "wrong" guess by resetting the entire game. Suppose the pursuer p is one level above the evader e. The evader picks at random one of the three vertices that are neighbors of e, located on the level below his current level. These always exist, except if the evader is at C_n. Observe that with probability 2/3, the evader and pursuer are no longer vertically aligned. If this happens, the evader can reset the game by moving to the top level (Section 6.2, Case (ii) b).

With the remaining probability of 1/3, the pursuer is still vertically aligned with the evader and the evader has got closer to C_n, resulting in a "correct" guess for the pursuer. To capture the evader, in the worst case, the pursuer needs n such consecutive guesses.

Next, we obtain a lower bound on the expected capture time.

The state of the game is measured by the capture time i.e. the expected number of steps required for the pursuer to capture the evader. The sequence of pursuer guesses and game outcomes can be modeled as a random walk on the directed graph shown in Figure 2. Here, the probability of progress (a "correct" guess by the pursuer) toward the capture state is 1/3 and the probability of the game being completely reset (a "wrong" guess by the pursuer) is 2/3.

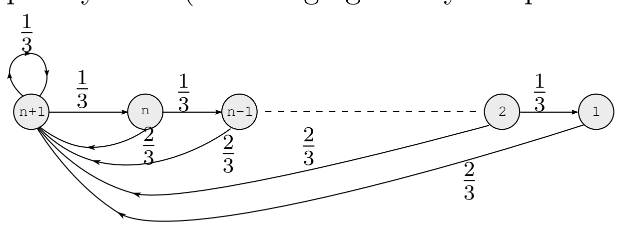

Fig. 2. A random walk on a line: The pursuer has to make consecutive correct guesses to prevent the game from being reset at any point in time.

The expected time it takes for this process to start at the left endpoint of the chain and arrive at the right endpoint is $\Omega(3^n)$ which gives us a lower bound on the expected duration of the game.

Theorem 8 *For every $n \in \mathbb{N}$, there exists an environment G_n with $4n + 1$ vertices and an evader strategy S such that the expected capture time of any pursuer strategy (for the zero-visibility pursuer) against S is $\Omega(3^n)$.*

6.4 Generalization to the k-visibility pursuer

We can obtain a lower bound on the capture time of a k-visibility pursuer using the same environment and the evader strategy S: suppose the pursuer starts at the apex and the evader is located at a vertex in C^{k+1}. Therefore, the pursuer can not see the evader. The evader follows S and resets the game if the pursuer makes a wrong guess – i.e. if the pursuer is no longer above the evader. Note that this requires entering pursuer's vision range. However, the pursuer still has to move to the apex to reach a configuration where he is above the evader (the evader will stop at C^{k+1}). On the other hand, if the pursuer reaches C^{n-k} by making correct guesses, the evader will be visible from that point on. Therefore, the pursuer does not need to make any further guesses.

To analyze the capture time, we can use a random walk on the line similar to the one shown in Figure 2. Since the pursuer stops guessing after reaching C^{n-k}, the length of the path is $n-k$ which gives us the following lower bound.

Theorem 9 *For every $n \in \mathbb{N}$, there exists an environment G_n with $4n + 1$ vertices and an evader strategy S such that the expected capture time of any pursuer strategy (for a k-visibility pursuer) against S is $\Omega(3^{n-k})$.*

Therefore, unless k is comparable to n, increasing the pursuer's visibility does not help much against a global visibility evader.

7 Symmetric Visibility

In this section, we study the cops and robbers game when both players have k-visibility. The case when $k = 0$ has been studied in [1] and $k = 1$ has been studied in [15]. In this section, we first show that the pursuer can establish eye-contact with the evader on any graph under the symmetric visibility model. That is, the pursuer can force the game into a configuration (p, e) such that p and e are visible from each other.

When studying the symmetric visibility game, the following question arises:

What is the class of graphs on which the evader can be captured with a single pursuer? The answer clearly depends on k. As mentioned earlier, when $k = 0$ the evader can be captured on any graph [1]. Further, the results in the previous sections imply that the evader can be caught on any distmantlable graph for arbitrary k. However, it is worth noting that it is possible to have situations where the evader will enter and leave the pursuer's sight until capture (an example is the game described in Section 6.4). It has also been shown that the class of graphs where the evader can be caught for $k = 1$ is larger than the class of dismantlable graphs [15].

In this section, we present a general result on establishing eye-contact in the symmetric visibility game. Next, we present a characterization of environments where a class of greedy algorithms suffices for capture (Section 8). On these environments, the pursuer can first establish eye contact and then follow the greedy strategy to capture the evader. Hence, this characterization yields a sufficient condition for capture in the symmetric visibility case. However, due to their widespread use [19,6,4], we believe that a characterization of environments where the greedy strategy works will be of independent interest as well.

We first show that the pursuer can establish eye-contact with the evader on any graph G for any k with high probability. The pursuer uses the following *strategy* S_p: Let n be the order of the graph G. The pursuer strategy consists of rounds of length n. At the beginning of the round, the pursuer is at some vertex which may be known by the evader. The pursuer picks a vertex (which we refer to as a destination) uniformly at random, moves to the destination via the shortest path, and waits there until the end of the round.

The basic idea is that, no matter what strategy the evader follows, the evader will be visible from a vertex v at the end of the round and the pursuer picks this vertex with probability $\frac{1}{n}$. The main technical challenge is the following. The pursuer picks v at the beginning of the round. If the evader can infer pursuer's choice using observations during the round, it may avoid being seen by avoiding the neighborhood of v. The following lemma shows that this can not happen. Namely, the evader can not obtain useful observation without risking being found by the pursuer.

Lemma 10 *On any graph connected G, and for any $k \geq 0$, a pursuer with k-visibility can move in such a way that, with high probability, no matter how an evader with k visibility moves, the game reaches a configuration (p, e) where e is visible from p.*

PROOF. Let S_p be the pursuer strategy described above. Suppose, for contradiction, that there exists an evader strategy S_e against S_p which guarantees that the probability of pursuer's finding the evader at the end of a round is zero. Since the evader will not see the pursuer during the execution of S_e, (i) the information it has about the location of the pursuer at the end of a round will be the same for all possible random destination choices of the pursuer, and thus (ii) we can assume that the evader makes its random choices a priori. Let π_e denote the evader's path while it is executing S_e during the round and w be the last vertex of π_e. With probability at least $\frac{1}{n}$, the pursuer will pick a neighbor of w as a destination and hence, find the evader at the end of the round. This contradicts the existence of S_e. $\square$

The expected time to find the evader using S_p is $O(n^2)$. It can be improved to $O(n \log n)$ by replacing S_p with the pursuer strategy in [1].

Note that for $k = 0$ the lemma implies capture. For $k = 1$, the result can be inferred from some of the results in [15].

8 Greedy strategies

Let (p, e) be the current configuration of a cops and robbers game on a graph G. The evader moves to a vertex $e' \in N(e)$ and suppose e' is visible from p. A natural pursuit strategy is to move toward e' on *a* shortest path from p to e'. Variants of such greedy strategies are extensively used in robotics because they are easy to implement and typically require little bookkeeping [19,6,4].

Two remarks are in order. First, observe that greedy strategies do not guarantee capture on some cop-win graphs. For example, in the environment of Section 6 (Figure 1), if the players are on the same cycle, a greedy pursuer will keep looping forever on this cycle. In contrast, an optimal strategy would move to the apex (which requires increasing the distance for some time) and then capture the evader.

Second, even though the greedy strategy is straightforward in Euclidean environments, it is not well-defined on graphs when there are multiple shortest paths. Let p and e be the vertices of the pursuer and evader respectively. Let $p_1, \ldots, p_q$ be neighbors of p that minimize the distance from e. Which vertex should the pursuer select? In this paper, we study two variants:

Class A Greedy Pursuit: In this version, we focus on a worst-case scenario and assume that the *evader* picks the shortest path the pursuer has to follow. The significance of this model is that if the pursuer can win in this model, then *any* choice of p_i yields capture.

Class B Greedy Pursuit: In this version, the *pursuer* is free to pick any shortest path between p and e'. If the pursuer can win in this model, this means that there is *some* choice of p_i that yields capture but additional reasoning (or perhaps randomization) is needed to find the right greedy strategy.

It is easy to see that if Class A pursuit succeeds, so will Class B pursuit. Also, note that in some environments (e.g. trees, Euclidean environments), the shortest path is unique. Hence, the distinction between Class A and B strategies disappear. Finally, there are environments where Class A pursuit fails but Class B pursuit succeeds. An example is shown in Figure 3.

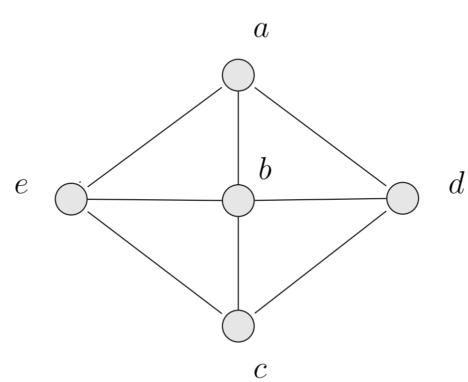

Fig. 3. On this graph Class A greedy pursuit fails but Class B greedy pursuit succeeds.

In Figure 3, suppose the pursuer is at a and the evader is at b. In Class A pursuit, the evader can move to c and force the pursuer to move to e. Afterward, the evader can make the pursuer chase the evader forever on the cycle $e \rightarrow c \rightarrow d \rightarrow a$. In contrast, a class B greedy pursuer can move to b and capture the evader.

8.1 *Class A Greedy Pursuit*

In this section, we use study the following greedy strategy. Let p and e be the locations of the pursuer and the evader when it is the pursuer's turn to move. Let $P = \{\pi_1, \ldots, \pi_k\}$ be the set of all shortest paths between p and e. Since the pursuer's choice of which shortest path to follow is arbitrary and we focus on the worst case scenario, we assume that the evader picks a path $\pi \in P$ from this set and the pursuer moves toward the evader on π.

We now present an algorithmic characterization (Algorithm Mark-Greedy) of graphs on which Class A greedy pursuit succeeds. In the following, $N(u, v)$ denotes the neighborhood of u with respect to v: $N(u, v) = \{w \in N(u) : d(w, v) = d(u, v) - 1\}$.

Theorem 11 *A graph G is greedy-cop-win if and only if Algorithm Mark-Greedy(G) marks all configurations of G.*

PROOF. Suppose that all configurations are marked. We show, by induction on the order of marked configurations, that the greedy pursuit strategy guarantees capture. The first configuration marked is of the form (v, v) for some vertex v. Therefore, if the game ever reaches this configuration, the evader is captured by definition. For inductive hypothesis, assume that greedy pursuit guarantees capture if the game ever enters one of the first k marked configurations. Consider the configuration (p, e) marked $(k + 1)^{th}$. It must be that, no matter which vertex e' the evader moves to, and no matter which shortest path it chooses for the pursuer, the pursuer will enter a configuration which was marked in the first k steps and win the game afterward.

Now suppose that after the execution of Mark-Greedy some configurations were unmarked. Let (p, e) be such a configuration and suppose the game starts at this configuration. It must be that there exists a vertex e' that the evader can move to, and a shortest path π between p and e' such that the resulting configuration (p', e') after the pursuer moves to p' on π is unmarked (Otherwise (p, e) would be marked.). Hence, the evader can guarantee that the game always remains in unmarked configurations. All capture configurations are of the form (v, v) and hence, marked. Therefore, the pursuer can never capture the evader with the greedy strategy. □

Algorithm Mark-Greedy(Graph $G = (V, E)$):
/* Initially, all configurations are unmarked */
Mark all configurations (v, v) for every vertex $v \in V$.
Repeat
Mark (p, e) if for all $e' \in N(e)$ and for all $p' \in N(p, e')$, (p', e') is marked.
Until no further marking is possible.

8.2 Class B Greedy Pursuit

Recall that the distinction between Class A and Class B pursuit is due to the choice of the shortest path the pursuer will follow. In Class B pursuit, the pursuer is free to choose any shortest path. Algorithm Mark-Greedy can be modified to recognize Class B greedy-cop-win graphs. The only modification needed is to replace the body of the loop with the line: Mark (p, e) if for all $e' \in N(e)$, *there exists* $p' \in N(p, e')$ such that (p', e') is marked. The proof of correctness is similar to Theorem 11 and omitted.

In addition to an algorithmic characterization, we show that Class B greedy pursuit succeeds on chordal graphs in the next section. However, there are non-chordal graphs where Class B pursuit succeeds as well. For example, the graph in Figure 3 has a chord-less cycle $(ecda)$.

8.2.1 Chordal graphs are Class B greedy-cop-win

A chord of a cycle is an edge connecting two non-consecutive vertices of a cycle. A graph G is chordal (or triangulated) if every cycle of length four or more contains a chord.

Chordal graphs form an important class of graphs which includes trees, cliques and interval graphs. In this section, we show that Class B greedy strategy works on chordal graphs.

Definition 12 *Let $C = \{v_1, \ldots, v_k\}$ be a cycle of a chordal graph G. Three vertices v_i, v_{i+1} and v_{i+2} form an ear of C if (v_i, v_{i+2}) is an edge.*

We will utilize the following lemma which is proved in the Appendix.

Lemma 13 *Every cycle C of a chordal graph has an ear.*

Since an ear is also a chord, Lemma 13 yields: A graph G is chordal if and only if every cycle of G has an ear.

Lemma 14 *Let G be a chordal graph. By following the Class B greedy strategy*

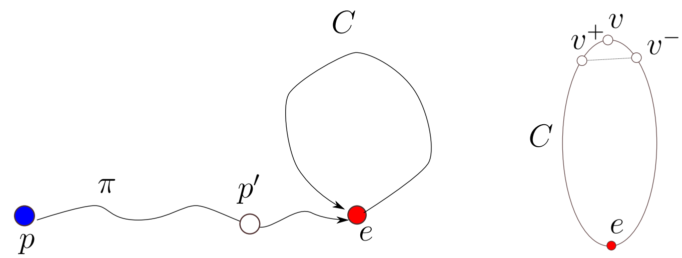

Fig. 4. The two cases that arise in Lemma 14.

described above, the pursuer can capture the evader on G.

PROOF. Suppose the pursuer and evader play the cops and robbers game on a graph G, starting from an arbitrary configuration. Let $d_1, d_2, \ldots, d_i, \ldots$ be a sequence where d_i is the distance between the players after the pursuer's i^{th} move. First, observe that when the pursuer is following the greedy strategy, no matter which shortest path is chosen, the distance between the pursuer and the evader after the pursuer's move never increases. We claim that if G is chordal, the pursuer can select the shortest paths in such a way that after a finite number of steps, the distance between the players decreases. Clearly, this implies capture. Let k be the current distance between the pursuer and the evader. For contradiction, assume that the claim is false. No matter what strategy the pursuer uses, the evader must revisit a vertex. Let e be the first revisited vertex and suppose the evader revisits e after l steps for the first time. Let C be the cycle formed by the evader's trajectory. Let p be the location of the pursuer when the evader is at e initially. Let π be a shortest path of length k between π and e. Note that, by assumption, the distance between the players remains k throughout these l steps. First suppose $k > l$. This case is shown in Figure 4-left. In this case, remaining on π is a valid Class B greedy strategy as it guarantees a separation of k. But then, when the evader revisits e, the distance between the players will decrease, leading to a contradiction. Therefore it must be that $k \leq l$. In this case, the pursuer can stay on π until e and follow C afterward. This also maintains a separation of k and hence it is a valid Class B strategy. Since the evader travels the full cycle C, it must go through a vertex v which, together with its neighbors v^- and v^+, form an ear (Figure 4-right). Let $H = (\pi + C) \subseteq G$. After the evader moves to v^+ and the pursuer completes his move on H, the distance between the two on H is k. However, since (v^-, v^+) is an edge, the distance between them on G is at most $k-1$. This means that the pursuer is not following a shortest path which is a contradiction. □

8.3 Symmetric visibility and greedy strategies

Consider the following model for symmetric visibility cops and robbers game. Both players have k-visibility, for some k. Suppose the game configuration is (p, e) with $d(p, e) = k$. Imagine the evader moves to e' with $d(p, e') = k + 1$. If we make the assumption that the pursuer can instantaneously observe the evader's motion and infer that the evader is on e', then it is easy to see that on greedy-cop-win graphs, the pursuer can capture the evader. First, it finds the evader (Lemma 10). Since the greedy pursuit guarantees that the distance never increases after the pursuer's move, the evader will be visible until the end of the game. If, however, the pursuer cannot observe the evader's motion instantaneously, the pursuer may not follow the greedy strategy. Recent results for greedy pursuit in convex planar environments under this model can be found in [6].

9 Conclusion

In this paper, we studied the effect of reducing the pursuer's visibility in the standard cops and robbers game. We showed that a pursuer with limited visibility is as powerful as a pursuer with global visibility in terms of the outcome of the game. On the other hand, we showed that the reduction in visibility can cause an exponential increase in the capture time. We also initiated the study of the cops and robbers game when the players have limited but symmetric visibility powers. For this version, we showed that the cop can establish eye contact with the robber on any graph. Next, we presented a characterization of graphs where a natural greedy strategy suffices for capture. This condition yields a sufficient condition for capture in the symmetric visibility case.

Our results shed light onto the role of information available to the pursuer on the outcome of this game and raises a number of interesting questions for future research:

- We have shown that a pursuer with limited visibility can still win on a cop-win graph but the capture time may increase exponentially. In applications where the capture time is crucial, the decrease in the sensing powers of the pursuers can be compensated by increasing the number of pursuers. What is a sufficient number of pursuers (with limited visibility) to make the capture time polynomial in the number of vertices?
- On a graph which is not cop-win, what is the number of pursuers (with limited visibility) sufficient to capture the evader?
- How does the outcome of the game change when the players have symmetric visibility? It seems unlikely that the exponential increase in the capture time

holds for this case.

Acknowledgment

This work is supported in part by NSF CCF-0634823 and NSF CNS-0707939. We thank the reviewers for useful comments. We also thank Joao Hespanha for useful discussions on Lemma 10.

APPENDIX

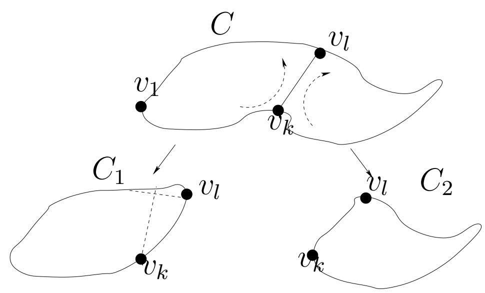

Fig. .1. Every cycle C in a chordal graph has an ear.

Proof of Lemma 13:

Let $C = v_1, \ldots, v_n$. We prove the lemma by induction on n. For the base case ($n = 3$), the lemma is trivially true. For the inductive step, assume that every cycle of length up to n in a chordal graph has an ear. Consider a cycle C with $|C| = n$. Since G is chordal, C must have a chord (v_k, v_l) for some k and l. Therefore we can divide C into two cycles C_1 and C_2. To obtain C_1, we start from v_k, follow the edge (v_k, v_l) and then C counter-clockwise. To obtain C_2, we follow C clockwise after v_l. We can rename the vertices so that $C_1 = \{v_1, \ldots v_k, v_l, v_{l+1}, \ldots, v_n\}$. By the inductive hypothesis, both C_1 and C_2 have ears. We focus on C_1. Two possible cases arise. In the first case, the ear is either $e_1 = (v_{k-1}, v_k, v_l)$ or $e_2 = (v_k, v_l, v_{l+1})$. If this is not the case, the ear of C_1 is also an ear in C and we are done. Therefore we focus on the first case. We form a new cycle C_1' as follows. If e_1 is the ear, $C_1' = \{v_1, \ldots v_{k-1}, v_l, \ldots, v_n\}$ (i.e. C_1' is obtained by deleting v_k from C_1). Otherwise, if e_2 is the ear, $C_1' = \{v_1, \ldots v_k, v_{l+1}, \ldots, v_n\}$ (i.e. v_l is deleted from C). We now apply the same argument to C_1'. Through this process, either we get to a point where C_1' has an ear which does not involve the chord or $|C_1'| = 3$, in which case, it will also have an ear which does not involve the chord. $\square$

References

[1] M. Adler, H. Räcke, N. Sivadasan, C. Sohler, and B. Vöcking. Randomized pursuit-evasion in graphs. *Proceedings of the International Colloquium on Automata, Languages and Programming (ICALP)*, pages 901–912, 2002.

[2] M. Aigner and M. Fromme. A game of cops and robbers. *Discrete Applied Math*, 8:1–12, 1984.

[3] R. Aleliunas, R. M. Karp, R. J. Lipton, L. Lovász, and C. Rackoff. Random walks, universal traversal sequences, and the complexity of maze problems. In *20th Annual Symposium on Foundations of Computer Science*, pages 218–223, San Juan, Puerto Rico, October 1979.

[4] S. Alexander, R. Bishop, and R. Ghrist. Pursuit and evasion in non-convex domains of arbitrary dimensions. In *Proceedings of Robotics: Science and Systems (to appear)*, Philadelphia, USA, August 2006.

[5] D. Bienstock. Graph searching, path-width, tree-width and related problems (a survey). *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 5:33–49, 1991.

[6] S. D. Bopardikar, F. Bullo, and J. P. Hespanha. Sensing limitations in the Lion and Man problem. In *ACC*, New York, July 2007. To appear.

[7] G. Brightwell and P. Winkler. Gibbs measures and dismantlable graphs. *J. Comb. Theory (Series B)*, 78:141–169, 2000.

[8] H. Cao, E. Ertin, V. Kulathumani, M. Sridharan, and A. Arora. Differential games in large-scale sensor-actuator networks. In *IPSN '06: Proceedings of the fifth international conference on Information processing in sensor networks*, pages 77–84, New York, NY, USA, 2006. ACM Press.

[9] I. Chatzigiannakis, S. Nikoletseas, and P. Spirakis. An efficient communication strategy for ad-hoc mobile networks. In *Proc. of 15th Symposium on Distributed Computing (DISC'2001)*, pages 285–299, 2001.

[10] J. A. Ellis, I. H. Sudborough, and J. S. Turner. The vertex separation and search number of a graph. *Information and Computation*, 113(1):50–79, 1994.

[11] S. Fitzpatrick and R. Nowakowski. Copnumber of graphs with strong isometric dimension two. *Ars combinatoria*, 59:65–73, 2001.

[12] A. S. Goldstein and E. M. Reingold. The complexity of pursuit on a graph. *Theor. Comput. Sci.*, 143(1):93–112, 1995.

[13] G. Hahn, F. Laviolette, N. Sauer, and R. Woodrow. On cop-win graphs. *Discrete Mathematics*, 258:27–41, 2002.

[14] G. Hahn and G. MacGillivray. A characterisation of k-cop-win graphs and digraphs. *Discrete Mathematics*, in press 2006.

[15] V. Isler, S. Kannan, and S. Khanna. Randomized pursuit-evasion with local visibility. *SIAM Journal on Discrete Mathematics*, 1(20):26–41, 2006. Earlier version in SODA 2004.

[16] S. Neufeld and R. Nowakowski. A game of cops and robbers played on products of graphs. *Discrete Mathematics*, 186:253–268, 1998.

[17] R. Nowakowski and P. Winkler. Vertex-to-vertex pursuit in a graph. *Discrete Math*, 43:235–239, 1983.

[18] A. Quilliot. *Homomorphismes, points fixes, rétractions et jeux de poursuite dans les graphes, les ensembles ordonnés et les espaces métriques*. PhD thesis, Universit de Paris VI, 1983.

[19] R. Vidal, O. Shakernia, J. Kim, D. Shim, and S. Sastry. Probabilistic pursuit-evasion games: Theory, implementation and experimental evaluation. *IEEE Transactions on Robotics and Automation*, 18:662–669, 2002.