The role of visibility in pursuit/evasion games

Athanasios Kehagias, Dieter Mitsche, Pawel Pralat

The Role of Visibility in Pursuit / Evasion Games

Athanasios Kehagias, Dieter Mitsche, and Paweł Prałat

February 20, 2014

Abstract

The *cops-and-robber* (CR) game has been used in mobile robotics as a discretized model (played on a graph G) of *pursuit/evasion* problems. The "classic" CR version is a *perfect information game*: the cops' (pursuer's) location is always known to the robber (evader) and vice versa. Many variants of the classic game can be defined: the robber can be *invisible* and also the robber can be either *adversarial* (tries to avoid capture) or *drunk* (performs a random walk). Furthermore, the cops and robber can reside in either nodes or edges of G. Several of these variants are relevant as models or robotic pursuit / evasion. In this paper, we first define carefully several of the variants mentioned above and related quantities such as the *cop number* and the *capture time*. Then we introduce and study the *cost of visibility* (COV), a quantitative measure of the increase in difficulty (from the cops' point of view) when the robber is invisible. In addition to our theoretical results, we present algorithms which can be used to compute capture times and COV of graphs which are analytically intractable. Finally, we present the results of applying these algorithms to the numerical computation of COV.

1 Introduction

Pursuit / evasion (PE) and related problems (search, tracking, surveillance) have been the subject of extensive research in the last fifty years and much of this research is connected to mobile robotics [7]. When the environment is represented by a graph[1], the original PE problem is reduced to a *graph game* played between the pursuers and the evader.

In the current paper, inspired by Isler and Karnad's recent work [21], we study the role of *information* in *cops-and-robber* (CR) games, an important version of graph-based PE. By "information" we mean specifically the players' *location*. For example, we expect that when the cops know the robber's location they can do better than when the robber is "invisible". Our goal is to make precise the term "better".

Reviews of the graph theoretic CR literature appear in [3, 5, 12]. In the "classical" CR variant [30] it is assumed that the cops always know the robber's location and vice versa. The "invisible" variant, in which the cops cannot see the robber (but the robber always sees the

[1]For instance, a floorplan can be modeled as a graph, with nodes corresponding to rooms and edges corresponding to doors. Similarly, a maze can be represented by a graph with edges corresponding to tunnels and nodes corresponding to intersections.

cops) has received less attention in the graph theoretic literature; among the few papers which treat this case we mention [20, 21, 22, 9] and also [1] in which *both* cops and robber are invisible.

Both the visible and invisible CR variants are natural models for discretized robotic PE problems; the connection has been noted and exploited relatively recently [20, 21, 38]. If it is further assumed that the robber is not actively trying to avoid capture (the case of *drunk* robber) we obtain a *one-player* graph game; this model has been used quite often in mobile robotics [13, 16, 17, 26, 35] and especially (when assuming random robber movement) in publications such as [19, 25, 32, 36, 37], which utilize *partially observable Markov decision processes* (POMDP, [15, 27, 29]). For a more general overview of pursuit/evasion and search problems in robotics, the reader is referred to [7]; some of the works cited in this paper provide a useful background to the current paper.

This paper is structured as follows. In Section 2 we present preliminary material, notation and the definition of the "classical" CR game; we also introduce several node and edge CR *variants*. In Section 3 we define rigorously the *cop number* and *capture time* for the classical CR game and the previously introduced CR variants. In Section 4 we study the *cost of visibility* (COV). In Section 5 we present algorithms which compute capture time and optimal strategies for several CR variants. In Section 6 we further study COV using computational experiments. Finally, in Section 7 we summarize and present our conclusions.

2 Preliminaries

2.1 Notation

1. We use the following notations for sets: $\mathbb{N}$ denotes $\{1, 2, \ldots\}$; $\mathbb{N}_0$ denotes $\{0, 1, 2, \ldots\}$; $[K]$ denotes $\{1, \ldots, K\}$; $A - B = \{x : x \in A, x \notin B\}$; $|A|$ denotes the *cardinality* of A (i.e., the number of its elements).

2. A *graph* $G = (V, E)$ consists of a *node set* V and an *edge set* E, where every $e \in E$ has the form $e = \{x, y\} \subseteq V$. In other words, we are concerned with finite, undirected, simple graphs; in addition we will always assume that G is connected and that G contains n nodes: $|V| = n$. Furthermore, we will assume, without loss of generality, that the node set is $V = \{1, 2, ..., n\}$. We let $V^K = \underbrace{V \times V \times \ldots \times V}_{K \text{ times}}$. We also define $V_D^2 \subseteq V^2$ by $V_D^2 = \{(x, x) : x \in V\}$ (it is the set of "diagonal" node pairs).

3. A *directed graph* (*digraph*) $G = (V, E)$ consists of a *node set* V and an *edge set* E, where every $e \in E$ has the form $e = (x, y) \in V \times V$. In other words, the edges of a digraph are *ordered* pairs.

4. In graphs, the (*open*) *neighborhood* of some $x \in V$ is $N(x) = \{y : \{x, y\} \in E\}$; in digraphs it is $N(x) = \{y : (x, y) \in E\}$. In both cases, the *closed neighborhood* of x is $N[x] = N(x) \cup \{x\}$.

5. Given a graph $G = (V, E)$, its *line graph* $L(G) = (V', E')$ is defined as follows: the node set is $V' = E$, i.e., it has one node for every edge of G; the edge set is defined by

having the nodes $\{u,v\},\{x,y\} \in V'$ connected by an edge $\{\{u,v\},\{x,y\}\}$ if and only if $|\{u,v\} \cap \{x,y\}| = 1$ (i.e., if the original edges of G are adjacent).

6. We will write $f(n) = o(g(n))$ if and only if $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$. Note that in this *asymptotic* notation n denotes the parameter with respect to which asymptotics are considered. So in later sections we will write $o(n)$, $o(M)$ etc.

2.2 The CR Game Family

The "classical" CR game can be described as follows. Player C controls K cops (with $K \geq 1$) and player R controls a single robber. Cops and robber are moved along the edges of a graph $G = (V, E)$ in discrete time steps $t \in \mathbb{N}_0$. At time t, the robber's location is $Y_t \in V$ and the cops' locations are $X_t = (X_t^1, X_t^2, \ldots, X_t^K) \in V^K$ (for $t \in \mathbb{N}_0$ and $k \in [K]$). The game is played in *turns*; in the 0-th turn first C places the cops on nodes of the graph and then R places the robber; in the t-th turn, for $t > 0$, *first* C moves the cops to X_t and *then* R moves the robber to Y_t. Two types of moves are allowed: (a) sliding along a single edge and (b) staying in place; in other words, for all t and k, either $\{X_{t-1}^k, X_t^k\} \in E$ or $X_{t-1}^k = X_t^k$; similarly, $\{Y_{t-1}, Y_t\} \in E$ or $Y_{t-1} = Y_t$. The cops win if they *capture* the robber, i.e., if there exist $t \in \mathbb{N}_0$ and $k \in [K]$ such that $Y_t = X_t^k$; the robber wins if for all $t \in \mathbb{N}_0$ and $k \in [K]$ we have $Y_t \neq X_t^k$. In what follows we will describe these eventualities by the following "shorthand notation": $Y_t \in X_t$ and $Y_t \notin X_t$ (i.e., in this notation we consider X_t as a *set* of cop positions).

In the classical game both C and R are *adversarial*: C plays to effect capture and R plays to avoid it. But there also exist "*drunk robber*" versions, in which the robber simply performs a *random walk* on G such that, for all $\forall u, v \in V$ we have

$$\Pr(Y_0 = u) = \frac{1}{n} \qquad \text{and} \qquad \Pr(Y_{t+1} = u | Y_t = v) = \begin{cases} \frac{1}{|N(v)|} & \text{if and only if } u \in N(v) \\ 0 & \text{otherwise} \end{cases} . \quad (1)$$

In this case we can say that no R player is present (or, following a common formulation, we can say that the R player is "Nature").

If an R player exists, the cops' locations are always known to him; on the other hand, the robber can be either *visible* (his location is known to C) or *invisible* (his location is unknown). Hence we have four different CR variants, as detailed in Table 1.

Adversarial Visible Robber	av-CR
Adversarial Invisible Robber	ai-CR
Drunk Visible Robber	dv-CR
Drunk Invisible Robber	di-CR

Table 1: Four variants of the CR game.

In all of the above CR variants both cops and robber move from node to node. This is a good model for entities (e.g., robots) which move from room to room in an indoor environment. There also exist cases (for example moving in a maze or a road network) where it makes more sense to assume that both cops and robber move from edge to edge. We will call the classical

version of the edge CR game *edge av-CR*; it has attracted attention only recently [10]. Edge ai-CR, dv-CR and di-CR variants are also possible, in analogy to the node versions listed in Table 1. Each of these cases can be reduced to the corresponding node variant, with the edge game taking place on the line graph $L(G)$ of G.

3 Cop Number and Capture Time

Two graph parameters which can be obtained from the av-CR game are the *cop number* and the *capture time*. In this section we will define these quantities in game theoretic terms [2] and also consider their extensions to other CR variants. Before examining each of these CR variants in detail, let us mention a particular modification which we will apply to all of them. Namely, we assume that (every variant of) the CR game is played for an *infinite number of rounds*. This is obviously the case if the robber is never captured; but we also assume that, in case the robber is captured at some time t^*, the game continues for $t \in \{t^*+1, t^*+2, \ldots\}$ with the following restriction: for all $t \geq t^*$, we have $Y_t = X_t^{k^*}$ (where k^* is the number of cop who effected the capture)[3].

3.1 The Node av-CR Game

We will define cop number and capture time in game theoretic terms. To this end we must first define *histories* and *strategies*.

A particular instance of the CR game can be fully described by the sequence of cops and robber locations; these locations are fully determined by the C and R moves. So, if we let $x_t \in V^K$ (resp. $y_t \in V$) denote the nodes into which C (resp. R) places the cops (resp. the robber) at time t, then a *history* is a sequence $x_0 y_0 x_1 y_1 \ldots$. Such a sequence can have finite or infinite length; we denote the set of all finite length histories by $H_*^{(K)}$; note that there exists an infinite number of finite length sequences. By convention $H_*^{(K)}$ also includes the *zero-length* or *null* history, which is the *empty sequence*[4], denoted by λ. Finally, we denote the set of all infinite length histories by $H_\infty^{(K)}$.

Since both cops and robber are visible and the players move sequentially, av-CR is a game of *perfect information*; in such a game C loses nothing by limiting himself to *pure* (i.e., deterministic) *strategies* [24]. A pure cop strategy is a function $s_C : H_*^{(K)} \to V^K$; a pure robber strategy is a function $s_R : H_*^{(K)} \to V$. In both cases the idea is that, given a finite length history, the strategy produces the next cop or robber move[5]; for example, when the robber strategy s_R receives the input x_0, it will produce the output $y_0 = s_R(x_0)$; when it receives $x_0 y_0 x_1$, it will produce $y_1 = s_R(x_0 y_0 x_1)$ and so on. We will denote the set of all legal cop strategies by $\mathbf{S}_C^{(K)}$ and the set of all legal robber strategies by $\mathbf{S}_R^{(K)}$; a strategy is "legal" if it only provides moves which

[2] While this approach is not common in the CR literature, we believe it offers certain advantages in clarity of presentation.

[3] This modification facilitates the game theoretic analysis presented in the sequel. Intuitively, it implies that after capture, the k^*-th cop forces the robber to "follow" him.

[4] This corresponds to the beginning of the game, when neither player has made a move, just before C places the cops on G.

[5] Note the dependence on K, the number of cops.

respect the CR game rules. The set $\widetilde{\mathbf{S}}_C^{(K)} \subseteq \mathbf{S}_C^{(K)}$ (resp. $\widetilde{\mathbf{S}}_R^{(K)} \subseteq \mathbf{S}_R^{(K)}$) is the set of *memoryless* legal cop (resp. robber) strategies, i.e., strategies which only depend only on the current cops and robber positions; we will denote the memoryless strategies by Greek letters, e.g., σ_C, σ_R etc. In other words

$$\sigma_C \in \widetilde{\mathbf{S}}_C^{(K)} \Rightarrow \left[\forall t : x_{t+1} = \sigma_C\left(x_0 y_0 \ldots . x_t y_t\right) = \sigma_C\left(x_t y_t\right)\right],$$
$$\sigma_R \in \widetilde{\mathbf{S}}_R^{(K)} \Rightarrow \left[\forall t : y_{t+1} = \sigma_R\left(x_0 y_0 \ldots . x_t y_t x_{t+1}\right) = \sigma_R\left(y_t x_{t+1}\right)\right].$$

It seems intuitively obvious that both C and R lose nothing by playing with memoryless strategies (i.e., computing their next moves based on the current position of the game, not on its entire history). This is true but requires a proof. One approach to this proof is furnished in [6, 14]. But we will present another proof by recognizing that the CR game belongs to the extensively researched family of *reachability games* [4, 28].

A reachability game is played by two players (Player 0 and Player 1) on a *digraph* $\overline{G} = \left(\overline{V}, \overline{E}\right)$; each node $v \in \overline{V}$ is a *position* and each edge is a *move*; i.e., the game moves from node to node (position) along the edges of the digraph. The game is described by the tuple $\left(\overline{V}_0, \overline{V}_1, \overline{E}, \overline{F}\right)$, where $\overline{V}_0 \cup \overline{V}_1 = \overline{V}$, $\overline{V}_0 \cap \overline{V}_1 = \emptyset$ and $\overline{F} \subseteq \overline{V}$. For $i \in \{0, 1\}$, $\overline{V}_i$ is the set of positions (nodes) from which the i-th Player makes the next move; the game terminates with a win for Player 0 if and only if a move takes place into a node $v \in \overline{F}$ (the *target set* of Player 0); if this never happens, Player 1 wins[6]. The following is well known [4, 28].

Theorem 3.1 *Let $\left(\overline{V}_0, \overline{V}_1, \overline{E}, \overline{F}\right)$ be a reachability game on the digraph $\overline{D} = \left(\overline{V}, \overline{E}\right)$. Then $\overline{V}$ can be partitioned into two sets $\overline{W}_0$ and $\overline{W}_1$ such that (for $i \in \{0, 1\}$) player i has a* memoryless *strategy σ_i which is winning whenever the game starts in $u \in \overline{W}_i$.*

We can convert the av-CR game with K cops to an equivalent reachability game which is played on the *CR game digraph*. In this digraph every node corresponds to a *position* of the original CR game; a (directed) edge from node u to node v indicates that it is possible to get from position u to position v in a single move. The CR game digraph has three types of nodes.

1. Nodes of the form $u = (x, y, p)$ correspond to positions (in the original CR game) with the cops located at $x \in V^K$, the robber at $y \in V$ and player $p \in \{C, R\}$ being next to move.

2. There is single node $u = (\lambda, \lambda, C)$ which corresponds to the starting position of the game: neither the cops nor the robber have been placed on G; it is C's turn to move (recall that λ denotes the empty sequence).

3. Finally, there exist n nodes of the form $u = (x, \lambda, R)$: the cops have just been placed in the graph (at positions $x \in V^K$) but the robber has not been placed yet; it is R's turn to move.

[6]Here is a more intuitive description of the game: each move consists in sliding a *token* from one digraph node to another, along an edge; the i-th player slides the token if and only if it is currently located on a node $v \in \overline{V}_i$ ($i \in \{0, 1\}$); Player 0 wins if and only if the token goes into a node $u \in \overline{F}$; otherwise Player 1 wins.

Let us now define

$$\begin{aligned}\overline{V}_0^{(K)} &= \left\{(x,y,C) : x \in V^K \cup \{\lambda\}, y \in V \cup \{\lambda\}\right\},\\ \overline{V}_1^{(K)} &= \left\{(x,y,R) : x \in V^K \cup \{\lambda\}, y \in V \cup \{\lambda\}\right\}\\ \overline{V}^{(K)} &= \overline{V}_0^{(K)} \cup \overline{V}_1^{(K)}\end{aligned}$$

and let $\overline{E}^{(K)}$ consist of all pairs (u,v) where $u,v \in \overline{V}^{(K)}$ and the move from u to v is legal. Finally, we recognize that C's target set is

$$\overline{F}^{(K)} = \left\{(x,y,p) : x \in V^K, y \in (V \cap x), p \in \{C,R\}\right\}.$$

i.e., the set of all positions in which the robber is in the same node as at least one cop.

With the above definitions, we have mapped the classical CR game (played with K cops on the graph G) to the reachability game $\left(\overline{V}_0^{(K)}, \overline{V}_1^{(K)}, \overline{E}^{(K)}, \overline{F}^{(K)}\right)$. By Theorem 3.1, Player i (with $i \in \{0,1\}$) will have a *winning set* $\overline{W}_i^{(K)} \subseteq \overline{V}^{(K)}$, i.e., a set with the following property: whenever the reachability game starts at some $u \in \overline{W}_i^{(K)}$, then Player i has a winning strategy (it may be the case, for specific G and K that either of $\overline{W}_0^{(K)}$, $\overline{W}_1^{(K)}$ is empty). Recall that in our formulation of CR as a reachability game, Player 0 is C. In reachability terms, the statement "C has a winning strategy in the classical CR game" translates to "$(\lambda,\lambda,C) \in \overline{W}_0^{(K)}$" and, for a given graph G, the validity of this statement will in general depend on K . It is clear that $\overline{W}_0^{(K)}$ is increasing with K:

$$K_1 \leq K_2 \Rightarrow \overline{W}_0^{(K_1)} \subseteq \overline{W}_0^{(K_2)}. \tag{2}$$

It is also also clear that

$$\text{"}(\lambda,\lambda,C) \in \overline{W}_0^{(|V|)}\text{" is true for every } G = (V,E), \tag{3}$$

because, if C has $|V|$ cops, he can place one in every $u \in V$ and win immediately[7].

Based on (2) and (3) we can define the *cop number* of G to be the minimum number of cops that guarantee capture; more precisely we have the following definition (which is equivalent to the "classical" definition of cop number [2]).

Definition 3.2 *The* cop number *of G is*

$$c(G) = \min\left\{K : (\lambda,\lambda,C) \in \overline{W}_0^{(K)}\right\}.$$

While a cop winning strategy s_C guarantees that the token will go into (and remain in) $\overline{F}^{(K)}$, we still do not know how long it will take for this to happen. However, it is easy to prove that, if $K \geq c(G)$ and C uses a *memoryless* winning strategy, then no game position will be repeated until capture takes place. Hence the following holds.

[7]In fact, for $K = |V|$, we have $\overline{W}_0^{(|V|)} = \overline{V}^{(K)}$, because from every position (x,y,p), C can move the cops so that one cop resides in each $u \in V$, which guarantees immediate capture.

Theorem 3.3 *For every G, let $K \geq c(G)$ and consider the CR game played on G with K cops. There exists a a memoryless cop winning strategy σ_C and a number $\overline{T}(K;G) < \infty$ such that, for every robber strategy s_R, C wins in no more than $\overline{T}(K;G)$ rounds.*

Let us now turn from winning to *time optimal* strategies. To define these, we first define the *capture time*, which will serve as the CR *payoff function*.

Definition 3.4 *Given a graph G, some $K \in \mathbb{N}$ and strategies $s_C \in \mathbf{S}_C^{(K)}$, $s_R \in \mathbf{S}_R^{(K)}$ the* av-CR capture time *is defined by*

$$T^{(K)}(s_C, s_R|G) = \min\left\{t : \exists k \in [K] \text{ such that } Y_t = X_t^k\right\}; \tag{4}$$

in case capture never takes place, we let $T^{(K)}(s_C, s_R|G) = \infty$.

We will assume that R's *payoff* is $T^{(K)}(s_C, s_R|G)$ and C's payoff is $-T^{(K)}(s_C, s_R|G)$ (hence av-CR is a *two-person zero-sum game*). Note that capture time (i) obviously depends on K and (ii) for a fixed K is fully determined by the s_C and s_R strategies. Now, following standard game theoretic practice, we define *optimal* strategies.

Definition 3.5 *For every graph G and $K \in \mathbb{N}$, the strategies $s_C^{(K)} \in \mathbf{S}_C^{(K)}$ and $s_R^{(K)} \in \mathbf{S}_R^{(K)}$ are a pair of* optimal strategies *if and only if*

$$\sup_{s_R \in \mathbf{S}_R^{(K)}} \inf_{s_C \in \mathbf{S}_C^{(K)}} T^{(K)}(s_C, s_R|G) = \inf_{s_C \in \mathbf{S}_C^{(K)}} \sup_{s_R \in \mathbf{S}_R^{(K)}} T^{(K)}(s_C, s_R|G). \tag{5}$$

The value *of the av-CR game played with K cops is the common value of the two sides of (5) and we denote it $T^{(K)}\left(s_C^{(K)}, s_R^{(K)}|G\right)$.*

We emphasize that the validity of (5) is not known *a priori*. C (resp. R) can guarantee that he loses no more than $\inf_{s_C \in \mathbf{S}_C^{(K)}} \sup_{s_R \in \mathbf{S}_R^{(K)}} T^{(K)}(s_C, s_R|G)$ (resp. gains no less than $\sup_{s_R \in \mathbf{S}_R^{(K)}} \inf_{s_C \in \mathbf{S}_C^{(K)}} T^{(K)}(s_C, s_R|G)$). We always have

$$\sup_{s_R \in \mathbf{S}_R^{(K)}} \inf_{s_C \in \mathbf{S}_C^{(K)}} T^{(K)}(s_C, s_R|G) \leq \inf_{s_C \in \mathbf{S}_C^{(K)}} \sup_{s_R \in \mathbf{S}_R^{(K)}} T^{(K)}(s_C, s_R|G). \tag{6}$$

But, since av-CR is an *infinite* game (i.e., depending on s_C and s_R, it can last an infinite number of turns) it is not clear that equality holds in (6) and, even when it does, the existence of optimal strategies $\left(s_C^{(K)}, s_R^{(K)}\right)$ which achieve the value is not guaranteed.

In fact it can be proved that, for $K \geq c(G)$, av-CR has both a value and optimal strategies. The details of this proof will be reported elsewhere, but the gist of the argument is the following. Since av-CR is played with $K \geq c(G)$ cops, by Theorem 3.3, C has a *memoryless* strategy which guarantees the game will last no more than $\overline{T}(K;G)$ turns. Hence av-CR with $K \geq c(G)$ essentially is a *finite* zero-sum two-player game; it is well known [31] that every such game has a value and optimal memoryless strategies. In short, we have the following.

Theorem 3.6 *Given any graph G and any $K \geq c(G)$, for the av-CR game there exists a pair $\left(\sigma_C^{(K)}, \sigma_R^{(K)}\right) \in \widetilde{\mathbf{S}}_C^{(K)} \times \widetilde{\mathbf{S}}_R^{(K)}$ of* memoryless time optimal strategies *such that*

$$T^{(K)}\left(\sigma_C^{(K)}, \sigma_R^{(K)}|G\right) = \sup_{s_R \in \mathbf{S}_R^{(K)}} \inf_{s_C \in \mathbf{S}_C^{(K)}} T^{(K)}(s_C, s_R|G) = \inf_{s_C \in \mathbf{S}_C^{(K)}} \sup_{s_R \in \mathbf{S}_R^{(K)}} T^{(K)}(s_C, s_R|G).$$

Hence we can define the capture time of a graph to be the value of av-CR when played on G with $K = c(G)$ cops.

Definition 3.7 *The* adversarial visible capture time *of G is*

$$ct(G) = \sup_{s_R \in \mathbf{S}_R^{(K)}} \inf_{s_C \in \mathbf{S}_C^{(K)}} T^{(K)}(s_C, s_R|G) = \inf_{s_C \in \mathbf{S}_C^{(K)}} \sup_{s_R \in \mathbf{S}_R^{(K)}} T^{(K)}(s_C, s_R|G).$$

with $K = c(G)$.

3.2 The Node dv-CR Game

In this game the robber is visible and performs a random walk on G (*drunk* robber) as indicated by (1). In the absence of cops, Y_t is a Markov chain on V, with transition probability matrix P, where for every $u, v \in \{1, 2, ..., |V|\}$ we have

$$P_{u,v} = \Pr(Y_{t+1} = u | Y_t = v).$$

In the presence of one or more cops, $\{Y_t\}_{t=0}^{\infty}$ is a *Markov decision process* (MDP) [33] with state space $V \cup \{n+1\}$ (where $n+1$ is the *capture state*) and transition probability matrix $P(X_t)$ (obtained from P as shown in [23]); in other words, X_t is the *control* variable, selected by C.

Since no robber strategy is involved, the capture time on G only depends on the (K-cops strategy) s_C: namely:

$$T^{(K)}(s_C|G) = \min\left\{t : \exists k \in [K] \text{ such that } Y_t = X_t^k\right\}, \tag{7}$$

which can also be written as

$$T^{(K)}(s_C|G) = \sum_{t=0}^{\infty} \mathbf{1}(Y_t \notin X_t), \tag{8}$$

where $\mathbf{1}(Y_t \notin X_t)$ equals 1 if Y_t does not belong to X_t (taken as a set of cop positions) and 0 otherwise. Since the robber performs a random walk on G, it follows that $T^{(K)}(s_C|G)$ is a random variable, and C wants to minimize its expected value:

$$E\left(T^{(K)}(s_C|G)\right) = E\left(\sum_{t=0}^{\infty} \mathbf{1}(Y_t \notin X_t)\right). \tag{9}$$

The minimization of (9) is a typical *undiscounted, infinite horizon* MDP problem. Using standard MDP results [33] we see that (i) C loses nothing by determining $X_0, X_1, \ldots$ through a memoryless strategy $\sigma_C(x, y)$ and (ii) for every $K \geq 1$, $E\left(T^{(K)}(\sigma_C|G)\right)$ is well defined. Furthermore, for every $K \in \mathbb{N}$ there exists an optimal strategy $\sigma_C^{(K)}$ which minimizes $E\left(T^{(K)}(\sigma_C|G)\right)$; hence we have the following.

Theorem 3.8 *Given any graph G and $K \in \mathbb{N}$, for the dv-CR game played on G with K cops there exists a memoryless strategy $\sigma_C^{(K)} \in \widetilde{\mathbf{S}}_C^{(K)}$ such that*

$$E\left(T^{(K)}\left(\sigma_C^{(K)}|G\right)\right) = \inf_{s_C \in \mathbf{S}_C^{(K)}} E\left(T^{(K)}\left(s_C|G\right)\right).$$

Definition 3.9 *The* drunk visible capture time *of G is*

$$dct\left(G\right) = \inf_{s_C \in \mathbf{S}_C^{(K)}} E\left(T^{(K)}\left(s_C|G\right)\right).$$

with $K = c\left(G\right)$.

Note that, even though a single cop suffices to capture the drunk robber on any G, we have chosen to define $dct\left(G\right)$ to be the capture time for $K = c\left(G\right)$ cops; we have done this to make (in Section 4) an equitable comparison between $ct\left(G\right)$ and $dct\left(G\right)$.

3.3 The Node ai-CR Game

This is *not* a perfect information game, since C cannot see R's moves. Hence C and R must use *mixed* strategies s_C, s_R. A mixed strategy s_C (resp. s_R) specifies, for every t, a conditional probability $\Pr\left(X_t|X_0, Y_0, \ldots, Y_{t-2}, X_{t-1}, Y_{t-1}\right)$ (resp. $\Pr\left(Y_t|X_0, Y_0, \ldots, Y_{t-1}, X_t\right)$) according to which C (resp. R) selects his t-th move. Let $\overline{\mathbf{S}}_C^{(K)}$ (resp. $\overline{\mathbf{S}}_R^{(K)}$) be the set of all mixed cop (resp. robber) strategies. A strategy pair $(s_R, s_C) \in \overline{\mathbf{S}}_C^{(K)} \times \overline{\mathbf{S}}_R^{(K)}$, specifies probabilities for all events $(X_0 = x_0, \ldots, X_t = x_t, Y_0 = y_0, \ldots, Y_t = y_t)$ and these induce a probability measure which in turn determines R's expected gain (and C's expected loss), namely $E\left(T^{(K)}\left(s_C^{(K)}, s_R^{(K)}|G\right)\right)$. Let us define

$$\underline{v}^{(K)} = \sup_{s_R \in \overline{\mathbf{S}}_R^{(K)}} \inf_{s_C \in \overline{\mathbf{S}}_C^{(K)}} E\left(T^{(K)}\left(s_C, s_R|G\right)\right),$$
$$\overline{v}^{(K)} = \inf_{s_C \in \overline{\mathbf{S}}_C^{(K)}} \sup_{s_R \in \overline{\mathbf{S}}_R^{(K)}} E\left(T^{(K)}\left(s_C, s_R|G\right)\right).$$

Similarly to av-CR, C (resp. R) can guarantee an expected payoff no greater than $\overline{v}^{(K)}$ (resp. no less than $\underline{v}^{(K)}$). If $\underline{v}^{(K)} = \overline{v}^{(K)}$, we denote the common value by $v^{(K)}$ and call it the *value* of the ai-CR game (played on G, with K cops). A pair of strategies $\left(s_C^{(K)}, s_R^{(K)}\right)$ is called optimal if and only if $E\left(T^{(K)}\left(s_C^{(K)}, s_R^{(K)}|G\right)\right) = v^{(K)}$.

In [22] we have studied the ai-CR game and proved that it does indeed have a value and optimal strategies. We give a summary of the relevant argument; proofs can be found in [22].

First, *invisibility does not increase the cop number*. In other words, there is a cop strategy (involving $c\left(G\right)$ cops) which guarantees bounded expected capture time for every robber strategy s_R. More precisely, we have proved the following.

Theorem 3.10 *On any graph G let $\overline{s}_C^{(K)}$ denote the strategy in which K cops random-walk on G. Then*

$$\forall K \geq c\left(G\right) : \sup_{s_R \in \overline{\mathbf{S}}_R^{(K)}} E\left(T^{(K)}\left(\overline{s}_C^{(K)}, s_R|G\right)\right) < \infty.$$

Now consider the "m-truncated" ai-CR game which is played exactly as the "regular" ai-CR but lasts at most m turns. Strategies $s_R \in \overline{\mathbf{S}}_R^{(K)}$ and $s_C \in \overline{\mathbf{S}}_C^{(K)}$ can be used in the m-truncated game: C and R use them only until the m-th turn. Let R receive one payoff unit for every turn in which the robber is not captured; denote the payoff of the m-truncated game (when strategies s_C, s_R are used) by $T_m^{(K)}(s_C, s_R|G)$. Clearly

$$\forall m \in \mathbb{N}, s_R \in \overline{\mathbf{S}}_R^{(K)}, s_C \in \overline{\mathbf{S}}_C^{(K)} : T_m^{(K)}(s_C, s_R|G) \leq T_{m+1}^{(K)}(s_C, s_R|G) \leq T^{(K)}(s_C, s_R|G)$$

The expected payoff of the m-truncated game is $E\left(T_m^{(K)}(s_C, s_R|G)\right)$. Because it is a *finite*, two-person, zero-sum game, the m-truncated game has a value and optimal strategies. Namely, the value is

$$v^{(K,m)} = \sup_{s_R \in \overline{\mathbf{S}}_R^{(K)}} \inf_{s_C \in \overline{\mathbf{S}}_C^{(K)}} E\left(T_m^{(K)}(s_C, s_R|G)\right) = \inf_{s_C \in \overline{\mathbf{S}}_C^{(K)}} \sup_{s_R \in \overline{\mathbf{S}}_R^{(K)}} E\left(T_m^{(K)}(s_C, s_R|G)\right)$$

and there exist optimal strategies $s_C^{(K,m)} \in \overline{\mathbf{S}}_C^{(K)}$, $s_R^{(K,m)} \in \overline{\mathbf{S}}_R^{(K)}$ such that

$$E\left(T_m^{(K)}\left(s_C^{(K,m)}, s_R^{(K,m)}|G\right)\right) = v^{(K,m)} < \infty. \tag{10}$$

In [22] we use the truncated games to prove that the "regular" ai-CR game has a value, an optimal C strategy and ε-optimal R strategies. More precisely, we prove the following.

Theorem 3.11 *Given any graph G and $K \geq c(G)$, the ai-CR game played on G with K cops has a value $v^{(K)}$ which satisfies*

$$\lim_{m\to\infty} v^{(K,m)} = \underline{v}^{(K)} = \overline{v}^{(K)} = v^{(K)}.$$

Furthermore, there exists a strategy $s_C^{(K)} \in \overline{\mathbf{S}}_C^{(K)}$ such that

$$\sup_{s_R \in \overline{\mathbf{S}}_R^{(K)}} E\left(T^{(K)}\left(s_C^{(K)}, s_R\right)\right) = v^{(K)}; \tag{11}$$

and for every $\varepsilon > 0$ there exists an m_ε and a strategy $s_R^{(K,\varepsilon)}$ such that

$$\forall m \geq m_\varepsilon : v^{(K)} - \varepsilon \leq \sup_{s_C \in \overline{\mathbf{S}}_C^{(K)}} E\left(T^{(K)}\left(s_C, s_R^{(K,\varepsilon)}\right)|G\right) \leq v^{(K)}. \tag{12}$$

Having established the existence of $v^{(K)}$ we have the following.

Definition 3.12 *The* adversarial invisible capture time *of G is*

$$ct_i(G) = v^{(K)} = \sup_{s_R \in \overline{\mathbf{S}}_R^{(K)}} \inf_{s_C \in \overline{\mathbf{S}}_C^{(K)}} E\left(T^{(K)}(s_C, s_R|G)\right) = \inf_{s_C \in \overline{\mathbf{S}}_C^{(K)}} \sup_{s_R \in \overline{\mathbf{S}}_R^{(K)}} E\left(T^{(K)}(s_C, s_R|G)\right)$$

with $K = c(G)$.

3.4 The Node di-CR Game

In this game Y_t is unobservable and drunk; call this the "regular" di-CR game and also introduce the m-truncated di-CR game. Both are one-player games or, equivalently, Y_t is a *partially observable MDP* (POMDP) [33]. The target function is

$$E\left(T^{(K)}\left(s_C|G\right)\right) = E\left(\sum_{t=0}^{\infty} \mathbf{1}\left(Y_t \notin X_t\right)\right), \tag{13}$$

which is exactly the same as (9) but now Y_t is *unobservable*. (13) can be approximated by

$$E\left(T_m^{(K)}\left(s_C|G\right)\right) = E\left(\sum_{t=0}^{m} \mathbf{1}\left(Y_t \notin X_t\right)\right). \tag{14}$$

The expected values in (13)-(14) are well defined for every s_C. C must select a strategy $s_C \in \overline{\mathbf{S}}_C^{(K)}$ which minimizes $E\left(T^{(K)}\left(s_C|G\right)\right)$. This is a typical *infinite horizon, undiscounted* POMDP problem [33] for which the following holds.

Theorem 3.13 *Given any graph G and $K \in \mathbb{N}$, for the di-CR game played on G with K cops there exists a strategy $s_C^{(K)} \in \overline{\mathbf{S}}_C^{(K)}$ such that*

$$E\left(T^{(K)}\left(s_C^{(K)}|G\right)\right) = \inf_{s_C \in \overline{\mathbf{S}}_C^{(K)}} E\left(T^{(K)}\left(s_C|G\right)\right).$$

Hence we can introduce the following.

Definition 3.14 *The* drunk invisible capture time *of G is*

$$dct_i\left(G\right) = \inf_{s_C \in \overline{\mathbf{S}}_C^{(K)}} E\left(T^{(K)}\left(s_C|G\right)\right).$$

with $K = c\left(G\right)$.

3.5 The Edge CR Games

As already mentioned, every edge CR variant can be reduced to the corresponding node variant played on $L\left(G\right)$, the line graph of G. Hence all the results and definitions of Sections 3.1 - 3.4 hold for the edge variants as well. In particular, we have an edge cop number $\overline{c}\left(G\right) = c\left(L\left(G\right)\right)$ and capture times

$$\overline{ct}\left(G\right) = ct\left(L\left(G\right)\right), \quad \overline{dct}\left(G\right) = dct\left(L\left(G\right)\right), \quad \overline{ct}_i\left(G\right) = ct_i\left(L\left(G\right)\right), \quad \overline{dct}_i\left(G\right) = dct_i\left(L\left(G\right)\right).$$

In general, all of these "edge CR parameters" will differ from the corresponding "node CR parameters".

4 The Cost of Visibility

4.1 Cost of Visibility in the Node CR Games

As already remarked, we expect that ai-CR is more difficult (from C's point of view) than av-CR (the same holds for the drunk counterparts of this game). We quantify this statement by introducing the *cost of visibility* (COV).

Definition 4.1 *For every G, the* adversarial cost of visibility *is $H_a(G) = \frac{ct_i(G)}{ct(G)}$ and the* drunk cost of visibility *is $H_d(G) = \frac{dct_i(G)}{dct(G)}$.*

Clearly, for every G we have $H_a(G) \geq 1$ and $H_d(G) \geq 1$ (i.e., it is at least as hard to capture an invisible robber than a visible one). The following theorem shows that in fact both $H_a(G)$ and $H_d(G)$ can become arbitrarily large. In proving the corresponding theorems, we will need the family of *long star graphs* $S_{N,M}$. For specific values of M and N, $S_{N,M}$ consists of N paths (we call these *rays*) each having M nodes, joined at a central node, as shown in Fig.1.

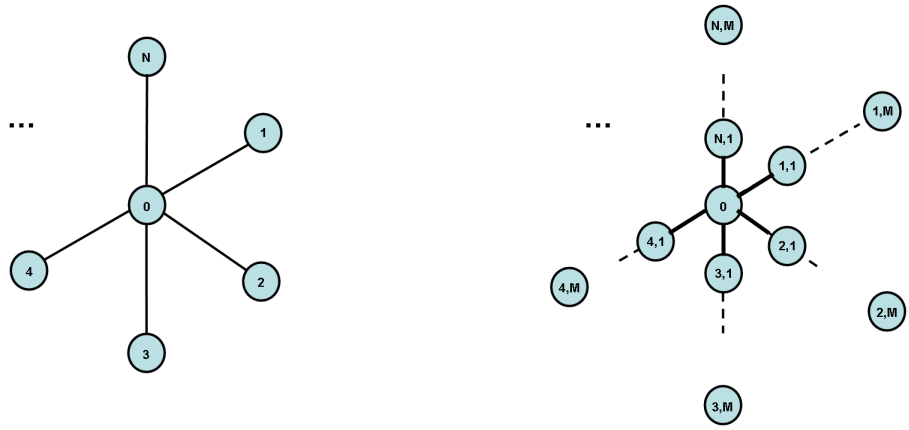

Figure 1: (a) The *star* graph $S_{N,1}$. (b) The *long star* graph $S_{N,M}$.

Theorem 4.2 *For every $N \in \mathbb{N}$ we have $H_a(S_{N,1}) = N$.*

Proof. **(i) Computing** $ct(S_{N,1})$. In av-CR, for every $N \in \mathbb{N}$ we have $ct(S_{N,1}) = 1$: the cop starts at $X_0 = 0$, the robber starts at some $Y_0 = u \neq 0$ and, at $t = 1$, he is captured by the cop moving into u; i.e., $ct(S_{N,1}) \leq 1$; on the other hand, since there are at least two vertices ($N \geq 1$), clearly $ct(S_{N,1}) \geq 1$.
(ii) Computing $ct_i(S_{N,1})$. Let us now show that in ai-CR we have $ct_i(S_{N,1}) = N$. C places the cop at $X_0 = 0$ and R places the robber at some $Y_0 = u \neq 0$. We will obtain $ct_i(S_{N,1})$ by bounding it from above and below. For an upper bound, consider the following C strategy. Since C does not know the robber's location, he must check the leaf nodes one by one. So at every odd t he moves the cop into some $u \in \{1, 2, \ldots, N\}$ and at every even t he returns to 0. Note that R cannot change the robber's original position; in order to do this, the robber must pass through 0 but then he will be captured by the cop (who either is already in 0 or will be moved into it just after the robber's move). Hence C can choose the nodes he will check on odd turns

with uniform probability and *without repetitions*. Equivalently, we can assume that the order in which nodes are chosen by C is selected uniformly at random from the set of all permutations; further, we assume that R (who does not know this order) starts at some $Y_0 = u \in \{1, \ldots, N\}$. Then we have

$$ct_i\left(S_{N,1}\right) \leq \frac{1}{N} \cdot 1 + \frac{1}{N} \cdot 3 + \ldots + \frac{1}{N} \cdot (2N-1) = N.$$

For a lower bound, consider the following R strategy. The robber is initially placed at a random leaf that is different than the one selected by C (if the cop did not start at the center). Knowing this, the best C strategy is to check (in any order) all leaves without repetition. If the cop starts at the center, we get exactly the same sum as for the upper bound. Otherwise, we have

$$ct_i\left(S_{N,1}\right) \geq \frac{1}{N-1} \cdot 2 + \frac{1}{N-1} \cdot 4 + \ldots + \frac{1}{N-1} \cdot (2N-2) = N.$$

(iii) Computing $H_a\left(S_{N,1}\right)$. Hence, for all $N \in \mathbb{N}$ we have $H_a\left(S_{N,1}\right) = \frac{ct_i\left(S_{N,1}\right)}{ct\left(S_{N,1}\right)} = N$. ■

Theorem 4.3 *For every $N \in \mathbb{N} - \{1\}$ we have*

$$H_d\left(S_{N,M}\right) = (1+o(1))\frac{(2N-1)(N-1)+1}{N} \geq 2N-3,$$

where the asymptotics is with respect to M; N is considered a fixed constant.

Proof. (i) Computing $dct\left(S_{N,M}\right)$. We will first show that, for any $N \in \mathbb{N}$, we have $dct\left(S_{N,M}\right) = \left(1+o\left(1\right)\right)\frac{M}{2}$ (recall that the parameter N is a fixed constant whereas $M \to \infty$.) Suppose that the cop starts on the i-th ray, at distance $(1+o(1))cM$ from the center (for some constant $c \in [0,1]$). The robber starts at a random vertex. It follows that for any j such that $1 \leq j \leq N$, the robber starts on the j-th ray with probability $(1+o(1))/N$. It is a straightforward application of Chernoff bounds[8] to show that with probability $1+o(1)$ the robber will not move by more than $o(M)$ in the next $O(MN) = O(M)$ steps, which suffice to finish the game. Hence, the expected capture time is easy to calculate.

- With probability $(1-c+o(1))/N$, the robber starts on the same ray as the cop but farther away from the center. Conditioning on this event, the expected capture time is $M(1-c+o(1))/2$.
- With probability $(c+o(1))/N$, the robber starts on the same ray as the cop but closer to the center. Conditioning on this event, the expected capture time is $M(c+o(1))/2$.
- With probability $(N-1+o(1))/N$, the robber starts on different ray than the cop. Conditioning on this event, the expected capture time is $(c+o(1))M + M(1/2+o(1))$.

[8] If X has a binomial distribution $Bin(n,p)$, then $Pr(|X-np| \geq \epsilon np) \leq 2\exp(-\epsilon^2 np/3)$ for any $\epsilon \leq 3/2$. Suppose the robber starts at distance $\omega(M^{2/3})$ from the center. During $N = O(M)$ steps the robber makes in expectation $N/2$ steps towards the center, and $N/2$ steps towards the end of the ray. The probability to make during N steps more than $N/2 + M^{2/3}$ steps towards the center, say, is thus at most $e^{-cM^{1/3}}$, and the same holds also by taking a union bound over all $O(M)$ steps. Hence, with probability at least $1 - e^{-cM^{1/3}}$ he will throughout $O(M)$ steps remain at distance $O(M^{2/3})$ from his initial position.

It follows that the expected capture time is

$$(1+o(1))M\left(\frac{1-c}{N}\cdot\frac{1-c}{2}+\frac{c}{N}\cdot\frac{c}{2}+\frac{N-1}{N}\cdot\frac{2c+1}{2}\right),$$

which is maximized for $c=0$, giving $dct\left(S_{N,M}\right)=\left(1+o\left(1\right)\right)\frac{M}{2}$.
(ii) Computing $dct_i\left(S_{N,M}\right)$. The initial placement for the robber is the same as in the visible variant, that is, the uniform distribution is used. However, since the robber is now invisible, C has to check all rays. As before, by Chernoff bounds, with probability at least $1-e^{-cM^{1/3}}$ (for some constant $c>0$) during $O(M)$ steps the robber is always within distance $O(M^{2/3})$ from its initial position. If the robber starts at distance $\omega(M^{2/3})$ from the center, he will thus with probability at least $1-e^{-cM^{1/3}}$ not change his ray during $O(M)$ steps. Otherwise, he might change from one ray to the other with bigger probability, but note that this happens only with the probability of the robber starting at distance $O(M^{2/3})$ from the center, and thus with probability at most $O(M^{-1/3})$. Keeping these remarks in mind, let us examine "reasonable" C strategies. It turns out there exist three such.
(ii.1) Suppose C starts at the end of one ray (chosen arbitrarily), goes to the center, and then successively checks the remaining rays without repetition, with probability at least $1-O(M^{-1/3})$, the robber will be caught. If the robber does not switch rays (and is therefore caught), the capture time is calculated as follows:

- With probability $(1+o(1))/N$, the robber starts on the same ray as the cop. Conditioning on this event, the expected capture time is $(1+o(1))M/2$.
- With probability $(1+o(1))/N$, the robber starts on the j-th ray visited by the cop. Conditioning on this event, the expected capture time is $(1+o(1))(M+2M(j-2)+M/2)$. (M steps are required to move from the end of the first ray to the center, $2M$ steps are 'wasted' to check $j-2$ rays, and $M/2$ steps are needed to catch the robber on the j-th ray, on expectation.)

Hence, conditioned under not switching rays, the expected capture time in this case is

$$\begin{aligned}
&(1+o(1))\frac{M}{N}\left(\frac{1}{2}+\left(1+\frac{1}{2}\right)+\left(3+\frac{1}{2}\right)+\ldots+\left(1+2(N-2)+\frac{1}{2}\right)\right)\\
&=(1+o(1))\frac{M}{N}\left(\frac{1}{2}+\left(2\cdot1-\frac{1}{2}\right)+\left(2\cdot2-\frac{1}{2}\right)+\ldots+\left(2(N-1)-\frac{1}{2}\right)\right)\\
&=(1+o(1))\frac{M}{N}\left(\frac{1}{2}+\frac{2N-1}{2}\cdot(N-1)\right)\\
&=(1+o(1))\frac{M}{2}\cdot\frac{(2N-1)(N-1)+1}{N}.
\end{aligned}$$

Otherwise, if the robber is not caught, C just randomly checks rays: starting from the center, C chooses a random ray, goes until the end of the ray, returns to the center, and continues like this, until the robber is caught. The expected capture time in this case is

$$\sum_{j\geq1}\left((1-\frac{1}{N})^{j-1}\frac{1}{N}\left(2(j-1)M+M/2\right)\right)=O(MN)=O(M).$$

Since this happens with probability $O(M^{-1/3})$, the contribution of the case where the robber switches rays is $o(M)$, and therefore for this strategy of C, the expected capture time is

$$(1+o(1))\frac{M}{2}\cdot\frac{(2N-1)(N-1)+1}{N}.$$

(ii.2) Now suppose C starts at the center of the ray, rather than the end, and checks all rays from there. By the same arguments as before, the capture time is

$$(1+o(1))\frac{M}{N}\left(\frac{1}{2}+\left(2+\frac{1}{2}\right)+\left(4+\frac{1}{2}\right)+\ldots+\left(2+2(N-2)+\frac{1}{2}\right)\right)$$

which is worse than in the case when starting at the end of a ray.
(ii.3) Similarly, suppose the cop starts at distance cM from the center, for some $c\in[0,1]$. If he first goes to the center of the ray, and then checks all rays (suppose the one he came from is the last to be checked), then the capture time is

$$(1+o(1))\frac{M}{N}\left(\frac{c^2}{2}+\left(c+\frac{1}{2}\right)+\left(c+2+\frac{1}{2}\right)+\ldots+\right.$$
$$\left.\left(\left(c+2(N-2)+\frac{1}{2}\right)+(1-c)\left(2c+2(N-1)+\frac{1-c}{2}\right)\right)\right),$$

which is minimized for $c=1$. And if C goes first to the end of the ray, and then to the center, the capture time is

$$(1+o(1))\frac{M}{N}\left(\frac{((1-c)^2}{2}+c\left(2(1-c)+\frac{c}{2}\right)+\left(2(1-c)+c+\frac{1}{2}\right)+\ldots+\right.$$
$$\left.\left(\left(2(1-c)+c+2(N-2)+\frac{1}{2}\right)\right),\right.$$

which for $N\geq 2$ is also minimized for $c=1$ (in fact, for $N=2$ the numbers are equal).

In short, the smallest capture time is achieved when C starts at the end of some ray and therefore

$$dct_i(S_{N,M})=(1+o(1))\frac{M}{2}\cdot\frac{(2N-1)(N-1)+1}{N}.$$

(iii) Computing $H_d(S_{N,M})$. It follows that for all $N\in\mathbb{N}-\{1\}$ we have

$$H_d(S_{N,M})=\frac{dct_i(S_{N,M})}{dct(S_{N,M})}=(1+o(1))\frac{(2N-1)(N-1)+1}{N}\geq 2N-3,$$

completing the proof. ■

4.2 Cost of Visibility in the Edge CR Games

The cost of visibility in the edge CR games is defined analogously to that of node games.

Definition 4.4 *For every G, the* edge adversarial cost of visibility *is* $\overline{H}_a(G)=\frac{\overline{ct}_i(G)}{\overline{ct}(G)}$ *and the* edge drunk cost of visibility *is defined as* $\overline{H}_d(G)=\frac{\overline{dct}_i(G)}{\overline{dct}(G)}$.

Clearly, for every G we have $\overline{H}_a(G) \geq 1$ and $\overline{H}_d(G) \geq 1$. The following theorems show that in fact both $\overline{H}_a(G)$ and $\overline{H}_d(G)$ can become arbitrarily large. To prove these theorems we will use the previously introduced star graph $S_{N,1}$ and its line graph which is the clique K_N. These graphs are illustrated in Figure 2 for $N = 6$.

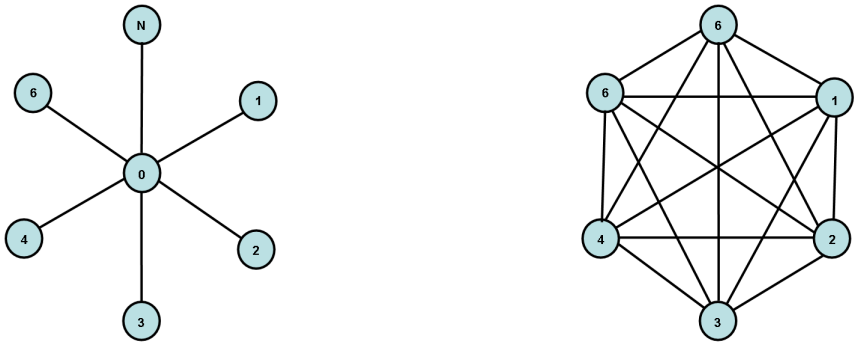

Figure 2: The *star* graph $S_{6,1}$ and its line graph, the clique K_6.

Theorem 4.5 *For every $N \in \mathbb{N} - \{1\}$ we have $\overline{H}_a(S_{N,1}) = N - 1$.*

Proof. We have $\overline{H}_a(S_{N,1}) = \frac{\overline{ct}_i(S_{N,1})}{\overline{ct}(S_{N,1})} = \frac{ct_i(K_N)}{ct(K_N)}$ and, since $N \geq 2$, clearly $ct(K_N) = 1$. Let us now compute $ct_i(K_N)$.

For an upper bound on $ct_i(K_N)$, C might just move to a random vertex. If the robber stays still or if he moves to a vertex different from the one occupied by C, he will be caught in the next step with probability $1/(N-1)$, and thus an upper bound on the capture time is $N - 1$.

For a lower bound, suppose that the robber always moves to a randomly chosen vertex, different from the one occupied by C, and including the one occupied by him now (that is, with probability $1/(N-1)$ he stands still, and after his turn, he is with probability $1/(N-1)$ at each vertex different from the vertex occupied by C. Hence C is forced to move, and since he has no idea where to go, the best strategy is also to move randomly, and the robber will be caught with probability $1/(N-1)$, yielding a lower bound on the capture time of $N - 1$. Therefore

$$ct_i(K_N) = N - 1,$$

Hence

$$\overline{H}_a(S_{N,1}) = \frac{\overline{ct}_i(S_{N,1})}{\overline{ct}(S_{N,1})} = \frac{ct_i(K_N)}{ct(K_N)} = N - 1.$$

■

Theorem 4.6 *For every $N \in \mathbb{N} - \{1\}$ we have $\overline{H}_d(S_{N,1}) = \frac{N(N-1)}{2N-3}$.*

Proof. This is quite similar to the adversarial case. We have $\overline{H}_d(S_{N,1}) = \frac{\overline{dct}_i(S_{N,1})}{\overline{dct}(S_{N,1})} = \frac{dct_i(K_N)}{dct(K_N)}$. Clearly we have $dct(K_N) = 1 - 1/N$ (with probability $1/N$ the robber selects the same vertex

to start with as the cop and is caught before the game actually starts; otherwise is caught in the first round).

For $dct_i(K_N)$, it is clear that the strategy of constantly moving is best for the cop, as in this case there are two chances to catch the robber (either by moving towards him, or by afterwards the robber moving onto the cop). It does not matter where he moves to as long as he keeps moving, and we may thus assume that he starts at some vertex v and moves to some other vertex w in the first round, then comes back to v and oscillates like that until the end of the game. When the cop moves to another vertex, the probability that the robber is there is $1/(N-1)$. If he is still not caught, the robber moves to a random place, thereby selecting the vertex occupied by the cop with probability $1/(N-1)$. Hence, the probability to catch the robber in one step is $\frac{1}{N-1} + (1 - \frac{1}{N-1})\frac{1}{N-1} = \frac{2N-3}{(N-1)^2}$. Thus, this time the capture time is a geometric random variable with probability of success equal to $\frac{2N-3}{(N-1)^2}$. We get $dct_i(K_N) = \frac{(N-1)^2}{2N-3}$ and so

$$\overline{H}_d(S_{N,1}) = \frac{\overline{dct}_i(S_{N,1})}{\overline{dct}(S_{N,1})} = \frac{dct_i(K_N)}{dct(K_N)} = \frac{(N-1)^2/(2N-3)}{(N-1)/N} = \frac{N(N-1)}{2N-3},$$

which can become arbitrarily large by appropriate choice of N. ■

5 Algorithms for COV Computation

For graphs of relatively simple structure (e.g., paths, cycles, full trees, grids) capture times and optimal strategies can be found by analytical arguments [22, 23]. For more complicated graphs, an algorithmic solution becomes necessary. In this section we present algorithms for the computation of capture time in the previously introduced node CR variants. The same algorithms can be applied to the edge variants by replacing G with $L(G)$.

5.1 Algorithms for Visible Robbers

5.1.1 Algorithm for Adversarial Robber

The av-CR capture time $ct(G)$ can be computed in polynomial time. In fact, stronger results have been presented by Hahn and MacGillivray; in [14] they present an algorithm which, given K, computes for every $(x, y) \in V^2$ the following:

1. $C(x, y)$, the optimal game duration when the cop/robber configuration is (x, y) and it is C's turn to play;

2. $R(x, y)$, the optimal game duration when the cop/robber configuration is (x, y) and it is R's turn to play.[9]

The av-CR capture time can be computed by $ct(G) = \min_{x \in V} \max_{y \in V} C(x, y)$; the optimal search strategies $\widehat{\sigma}_C$, $\widehat{\sigma}_R$ can also be easily obtained from the *optimality equations*, as will be seen a little later. We have presented in [23] an implementation of Hahn and MacGillivray's algorithm, which we call CAAR (Cops Against Adversarial Robber). Below we present the algorithm for the case of a single cop (the generalization for more than one cop is straightforward).

[9]When $K < c(G)$, there exist (x, y) such that $C(x, y) = R(x, y) = \infty$; Hahn and MacGillivray's algorithm computes this correctly, as well.

The Cops Against Adversarial Robber (CAAR) Algorithm

Input: $G = (V, E)$
01 `For All` $(x, y) \in V_D^2$
02 $C^{(0)}(x, y) = 0$
03 $R^{(0)}(x, y) = 0$
04 `EndFor`
05 `For All` $(x, y) \in V^2 - V_D^2$
06 $C^{(0)}(x, y) = \infty$
07 $R^{(0)}(x, y) = \infty$
08 `EndFor`
09 $i = 1$
10 `While` $1 > 0$
11 `For All` $(x, y) \in V^2 - V_D^2$
12 $C^{(i)}(x, y) = 1 + \min_{x' \in N[x]} R^{(i-1)}(x', y)$
13 $R^{(i)}(x, y) = 1 + \max_{y' \in N[y]} C^{(i)}(x, y')$
14 `EndFor`
15 `If` $C^{(i)} = C^{(i-1)}$ `And` $R^{(i)} = R^{(i-1)}$
16 `Break`
17 `EndIf`
18 $i \leftarrow i + 1$
19 `EndWhile`
20 $C = C^{(i)}$
21 $R = R^{(i)}$
Output: C, R

The algorithm operates as follows. In lines `01-08` $C^{(0)}(x, y)$ and $R^{(0)}(x, y)$ are initialized to ∞, except for "diagonal" positions $(x, y) \in V_D^2$ (i.e., positions with $x = y$) for which we obviously have $C(x, x) = R(x, x) = 0$. Then a loop is entered (lines `10-19`) in which $C^{(i)}(x, y)$ is computed (line `12`) by letting the cop move to the position which achieves the smallest capture time (according to the currently available estimate $R^{(i-1)}(x, y)$); $R^{(i)}(x, y)$ is computed similarly in line `13`, looking for the largest capture time. This process is repeated until no further changes take place, at which point the algorithm exits the loop and terminates[10]. It has been proved in [14] that, for any graph G and any $K \in \mathbb{N}$, CAAR always terminates and the finally obtained (C, R) pair satisfies the *optimality equations*

$$\forall (x, y) \in V_D^2 : C(x, y) = 0; \quad \forall (x, y) \in V^2 - V_D^2 : C(x, y) = 1 + \min_{x' \in N[x]} R(x', y), \tag{15}$$

$$\forall (x, y) \in V_D^2 : R(x, y) = 0; \quad \forall (x, y) \in V^2 - V_D^2 : R(x, y) = 1 + \max_{y' \in N[y]} C(x, y'). \tag{16}$$

[10]This algorithm is a game theoretic version of *value iteration* [33], which we see again in Section 5.2.

The optimal memoryless strategies $\sigma_C^{(K)}(x,y)$, $\sigma_R^{(K)}(x,y)$ can be computed for every position (x,y) by letting $\sigma_C^{(K)}(x,y)$ (resp. $\sigma_R^{(K)}(x,y)$) be a node $x' \in N[x]$ (resp. $y' \in N[y]$) which achieves the minimum in (15) (resp. maximum in (16)). The capture time $ct(G)$ is computed from

$$ct(G) = \min_{x \in V} \max_{y \in V} C(x,y).$$

5.1.2 Algorithm for Drunk Robber

For any given K, *value iteration* can be used to determine both $dct(G,K)$ and the optimal strategy $\sigma_C^{(K)}(x,y)$; one implementation is our CADR (Cops Against Drunk Robber) algorithm [23] which is a typical value-iteration [33] MDP algorithm; alternatively, CADR can be seen as an extension of the CAAR idea to the dv-CR. Below we present the algorithm for the case of a single cop (the generalization for more than one cops is straightforward).

The Cops Against Drunk Robber (CADR) Algorithm

Input: $G = (V,E)$, ε
01 `For All` $(x,y) \in V_D^2$
02 $C^{(0)}(x,y) = 0$
03 `EndFor`
04 `For All` $(x,y) \in V - V_D^2$
05 $C^{(0)}(x,y) = \infty$
06 `EndFor`
07 $i = 1$
08 `While` $1 > 0$
09 `For All` $(x,y) \in V - V_D^2$
10 $C^{(i)}(x,y) = 1 + \min_{x' \in N[x]} \sum_{y' \in V} P((x',y) \to (x',y')) C^{(i-1)}(x',y')$
11 `EndFor`
12 `If` $\max_{(x,y) \in V^2} \left| C^{(i)}(x,y) - C^{(i-1)}(x,y) \right| < \varepsilon$
13 `Break`
14 `EndIf`
15 $i \leftarrow i + 1$
16 `EndWhile`
17 $C = C^{(i)}$
Output: C

The algorithm operates as follows (again we use $C(x,y)$ to denote the optimal expected game duration when the game position is (x,y)). In lines `01-06` $C^{(0)}(x,y)$ is initialized to ∞, except for "diagonal"positions $(x,y) \in V_D^2$. In the main loop (lines `08-16`) $C^{(i)}(x,y)$ is computed (line `10`) by letting the cop move to the position which achieves the smallest expected capture time ($P((x',y) \to (x',y'))$ in line `10` indicates the transition probability from(x',y) to (x',y')). This process is repeated until the maximum change $\left| C^{(i)}(x,y) - C^{(i-1)}(x,y) \right|$ is smaller than

the *termination criterion* ε, at which point the algorithm exits the loop and terminates. This is a typical *value iteration* MDP algorithm [33]; the convergence of such algorithms has been studied by several authors, in various degrees of generality [8, 11, 18]. A simple yet strong result, derived in [11], uses the concept of *proper* strategy: a strategy is called proper if it yields finite expected capture time. It is proved in [11] that, if a proper strategy exists for graph G, then CADR-like algorithms converge. In the case of dv-CR we know that C has a proper strategy: it is the random walking strategy $\overline{s}_C^{(K)}$ mentioned in Theorem 3.10. Hence CADR converges and in the limit, $C = \lim_{i\to\infty} C^{(i)}$ satisfies the *optimality equations*

$$\forall (x,y) \in V_D^2 : C(x,y) = 0; \quad \forall (x,y) \in V^2 - V_D^2 : C(x,y) = 1 + \min_{x' \in N[x]} \sum P((x',y) \to (x',y'))\, C(x',y'). \tag{17}$$

The optimal memoryless strategy $\sigma_C^{(K)}(x,y)$ can be computed for every position (x,y) by letting $\sigma_C^{(K)}(x,y)$ be a node $x' \in N[x]$ (resp. $y' \in N[y]$) which achieves the minimum in (15) (resp. maximum in (16)). The capture time $dct(G)$ is computed from

$$dct(G) = \min_{x \in V} C(x,y).$$

5.2 Algorithms for Invisible Robbers

5.2.1 Algorithms for Adversarial Robber

We have not been able to find an efficient algorithm for solving the ai-CR game. Several algorithms for *imperfect information stochastic games* could be used to this end but we have found that they are practical only for very small graphs.

5.2.2 Algorithm for Drunk Robber

In the case of the drunk invisible robber we are also using a game tree search algorithm with pruning, for which some analytical justification can be provided. We call this the *Pruned Cop Search* (PCS) algorithm. Before presenting the algorithm we will introduce some notation and then prove a simple fact about expected capture time. We limit ourselves to the single cop case, since the extension to more cops is straightforward.

We let $\mathbf{x} = x_0 x_1 x_2 \ldots$ be an infinite history of cop moves. Letting t being the current time step, the probability vector $\mathbf{p}(t)$ contains the probabilities of the robber being in node $v \in V$ or in the *capture state* $n+1$; more specifically: $\mathbf{p}(t) = [p_1(t), \ldots, p_v(t), \ldots, p_n(t), p_{n+1}(t)]$ and $p_v(t) = \Pr(y_t = v | x_0 x_1 \ldots x_t)$. Hence $\mathbf{p}(t)$ depends (as expected) on the *finite* cop history $x_0 x_1 \ldots x_t$. The expected capture time is denoted by $C(\mathbf{x}) = E(T|\mathbf{x})$; the conditioning is on the infinite cop history. The PCS algorithm works because $E(T|\mathbf{x})$ can be approximated from a finite part of $\mathbf{x}$, as explained below. We have

$$C(\mathbf{x}) = E(T|\mathbf{x}) = \sum_{t=0}^{\infty} t \cdot \Pr(T = t|\mathbf{x}) = \sum_{t=0}^{\infty} \Pr(T > t|\mathbf{x}). \tag{18}$$

$\mathbf{x}$ in the conditioning is the *infinite* history $\mathbf{x} = x_0 x_1 x_2 \ldots$. However, for every t we have

$$\Pr(T > t|\mathbf{x}) = 1 - \Pr(T \le t|\mathbf{x}) = 1 - \Pr(T \le t|x_0 x_1 \ldots x_t).$$

Let us define

$$C^{(t)}(x_0x_1\dots x_t) = \sum_{\tau=0}^{t}\left[1-\Pr\left(T\le\tau|x_0x_1\dots x_\tau\right)\right] = \sum_{\tau=0}^{t}\left[1-p_{n+1}(\tau)\right],$$

where $p_{n+1}(\tau)$ is the probability that the robber is in the *capture state* $n+1$ at time τ (the dependence on $x_0x_1\dots x_\tau$ is suppressed, for simplicity of notation). Then for all t we have

$$C^{(t)}(x_0x_1\dots x_t) = C^{(t-1)}(x_0x_1\dots x_{t-1}) + (1-p_{n+1}(t)). \tag{19}$$

Update (19) can be computed using only the previous cost $C^{(t-1)}(x_0x_1\dots x_{\tau-1})$ and the (previously computed) probability vector $\mathbf{p}(t)$. While $C^{(t)}(x_0\dots x_t)\le C(\mathbf{x})$, we hope that (at least for the "good" histories) we have

$$\lim_{t\to\infty} C^{(t)}(x_0\dots x_t) = C(\mathbf{x}). \tag{20}$$

This actually works well in practice.

The PCS algorithm is given below in pseudocode. We have introduced a structure S with fields $S.\mathbf{x}$, $S.\mathbf{p}$, $S.C = C(S.\mathbf{x})$. Also we denote concatenation by the | symbol, i.e., $x_0x_1\dots x_t|v = x_0x_1\dots x_tv$.

The Pruned Cop Search (PCS) Algorithm

Input: $G=(V,E)$, x_0, J_{max}, ε

```
01  t = 0
02  S.x = x_0, S.p = Pr(y_0|x_0), S.C = 0
03  S = {S}
04  C_best^old = 0
05  While 1 > 0
06        S~ = ∅
07        For All S ∈ S
08            x = S.x, p = S.p, C = S.C
09            For All v ∈ N[x_t]
10                x' = x|v
11                p' = p · P(v)
12                C' = Cost(x', p', C)
13                S'.x = x', S'.p = p', S'.C = C'
14                S~ = S~ ∪ {S'}
15            EndFor
16        EndFor
17        S = Prune(S~, J_max)
18        [x_best, C_best] = Best(S)
19        If |C_best − C_best^old| < ε
20            Break
21        Else
```

22 $C_{best}^{old} = C_{best}$
23 $t \leftarrow t+1$
24 EndIf
25 EndWhile
Output: $\mathbf{x}_{best}$, $C_{best} = C(\mathbf{x}_{best})$.

The PCS algorithm operates as follows. At initialization (lines 01-04), we create a single S structure (with $S.\mathbf{x}$ being the initial cop position, $S.\mathbf{p}$ the initial, uniform robber probability and $S.C = 0$) which we store in the set $\mathbf{S}$. Then we enter the main loop (lines 05-25) where we pick each available cop sequence $\mathbf{x}$ of length t (line 08). Then, in lines 09-15 we compute, for all legal extensions $\mathbf{x}' = \mathbf{x}|v$ (where $v \in N[x_t]$) of length $t+1$ (line 10), the corresponding $\mathbf{p}'$ (line 11) and C' (by the subroutine **Cost** at line 12). We store these quantities in S' which is placed in the temporary storage set $\widetilde{\mathbf{S}}$ (lines 13-14). After exhausting all possible extensions of length $t+1$, we prune the temporary set $\widetilde{\mathbf{S}}$, retaining only the $J_{\max}$ best cop sequences (this is done in line 17 by the subroutine **Prune** which computes "best" in terms of smallest $C(\mathbf{x})$). Finally, the subroutine **Best** in line 18 computes the overall smallest expected capture time $C_{best} = C(\mathbf{x}_{best})$. The procedure is repeated until the termination criterion $|C_{best} - C_{best}^{old}| < \varepsilon$ is satisfied. As explained above, the criterion is expected to be always eventually satisfied because of (20).

6 Experimental Estimation of The Cost of Visibility

We now present numerical computations of the drunk cost of visibility for graphs which are not amenable to analytical computation[11]. In Section 6.1 we deal with node games and in Section 6.2 with edge games.

6.1 Experiments with Node Games

Since $H_d(G) = \frac{dct_i(G)}{dct(G)}$, we use the CADR algorithm to compute $dct(G)$ and the PCS algorithm to compute $dct_i(G)$. We use graphs G obtained from *indoor environments*, which we represent by their *floorplans*. In Fig. 3 we present a floorplan and its graph representation. The graph is obtained by decomposing the floorplan into convex *cells*, assigning each cell to a node and connecting nodes by edges whenever the corresponding cells are connected by an open space.

We have written a script which, given some parameters, generates random floorplans and their graphs. Every floorplan consists of a rectangle divided into orthogonal "rooms". If each internal room were connected to its four nearest neighbors we would get an $M \times N$ grid G'. However, we randomly generate a spanning tree G_T of G' and initially introduce doors only

[11] We do not deal with the adversarial cost of visibility because, while we can compute $ct(G)$ with the CAAR algorithm, we do not have an efficient algorithm to compute $ct_i(G)$; hence we cannot perform experiments on $H_a(G) = \frac{ct_i(G)}{ct(G)}$. The difficulty with $ct_i(G)$ is that ai-CR is a stochastic game of imperfect information; even for very small graphs, one cop and one robber, ai-CR involves a state space with size far beyond the capabilities of currently available stochastic games algorithms (see [34]).

between rooms which are connected in G_T. Our final graph G is obtained from G_T by iterating over all missing edges and adding each one with probability $p_0 \in [0, 1]$. Hence each floorplan is characterized by three parameters: M, N and p_0.

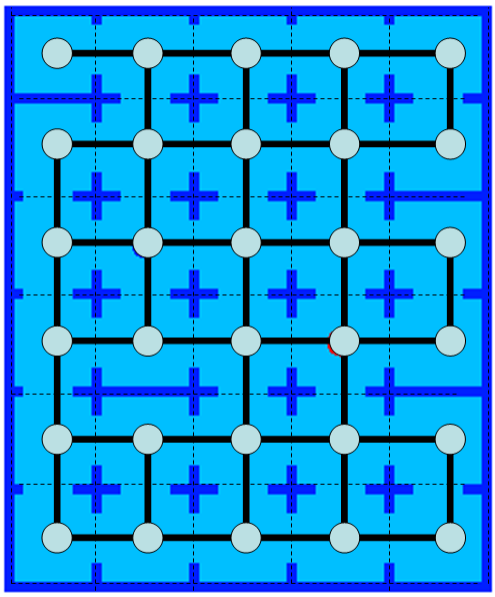

Figure 3: A floorplan and the corresponding graph.

We use the following pairs of (M, N) values: (1,30), (2,15), (3,10), (4,7), (5,6). Four of these pairs give a total of 30 nodes and the pair ($M = 4$, $N = 7$) gives $n = 28$ nodes; as M/N increases, we progress from a path to a nearly square grid. For each (M, N) pair we use five p_0 values: 0.00, 0.25, 0.50, 0.75, 1.00; note the progression from a tree ($p_0 = 0.00$) to a full grid ($p_0 = 1.00$). For each triple (M, N, p_0) we generate 50 floorplans, obtain their graphs and for each graph G we compute $dct(G)$ using CADR, $dct_i(G)$ using PCS and $H_d(G) = \frac{dct_i(G)}{dct(G)}$; finally we average $H_d(G)$ over the 100 graphs. In Fig. 4 we plot $dct(G)$ as a function of the probability p_0; each plotted curve corresponds to an (M, N) pair. Similarly, in Fig. 5 we plot $dct_i(G)$ and in Fig. 6 we plot $H_d(G)$.

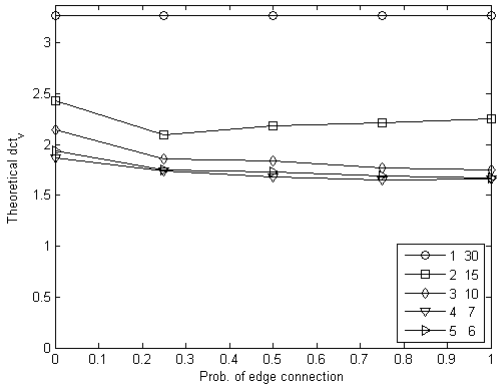

Figure 4: $dct(G)$ curves for floorplans with n=30 or n=28 cells. Each curve corresponds to a fixed (M, N) pair. The horizontal axis corresponds to the edge insertion probability p_0.

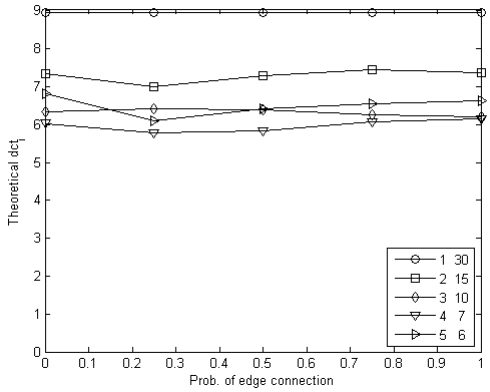

Figure 5: $dct_i(G)$ curves for floorplans with n=30 or n=28 cells. Each curve corresponds to a fixed (M, N) pair. The horizontal axis corresponds to the edge insertion probability p_0.

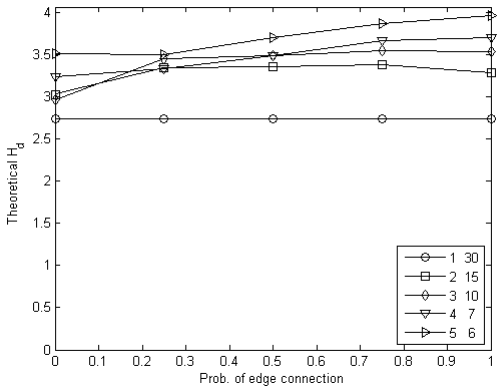

Figure 6: $H_d(G)$ curves for floorplans with n=30 or n=28 cells. Each curve corresponds to a fixed (M, N) pair. The horizontal axis corresponds to the edge insertion probability p_0.

We can see in Fig. 4-5 that both $dct\,(G)$ and $dct_i\,(G)$ are usually decreasing functions of the M/N ratio. However the cost of visibility $H_d\,(G)$ *increases* with M/N. This is due to the fact that, when the M/N ratio is low, G is closer to a path and there is less difference in the search schedules and capture times between dv-CR and di-CR. On the other hand, for high M/N ratio, G is closer to a grid, with a significantly increased ratio of edges to nodes (as compared to the low M/N, path-like instances). This, combined with the loss of information (visibility), results in $H_d(G)$ being an increasing function of M/N. The increase of $H_d\,(G)$ with p_0 can be explained in the same way, since increasing p_0 implies more edges and this makes the cops' task harder.

6.2 Experiments with Edge Games

Next we deal with $\overline{H}_d\,(G) = \frac{\overline{dct}_i(G)}{\overline{dct}(G)}$. We use graphs G obtained from *mazes* such as the one illustrated in Fig. 7. Every *corridor* of the maze corresponds to an edge; corridor intersections correspond to nodes. The resulting graph G is also depicted in Fig.7. From G we obtain the line graph $L(G)$, to which we apply CADR to compute $dct\,(L(G)) = \overline{dct}\,(G)$ and PCS to compute $dct_i\,(L(G)) = \overline{dct}_i\,(G)$.

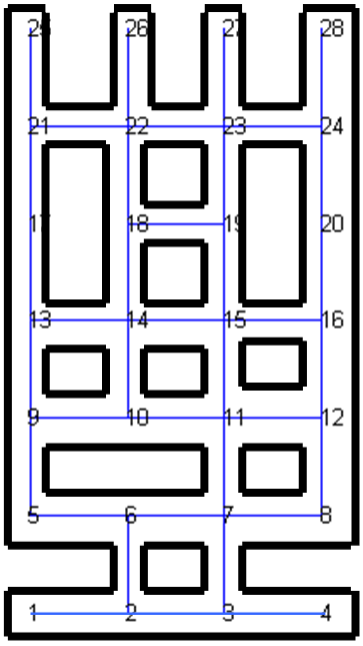

Figure 7: A maze and the corresponding graph.

We use graphs of the same type as the ones of Section 6.1 but we now focus on the edge-to-edge movements of cops and robber. Hence from every G (obtained by a specific (M, N, p_0) triple) we produce the line graph $L(G)$, for which we compute $H_d(L(G))$ using the CADR and PCS algorithms. Once again we generate 50 graphs and present average $dct(G)$, $dct_i\,(G)$ and $H_d\,(G)$ results in Figures 8-10. These figures are rather similar to Figures 4-6, except that the increase of $\overline{H}_d\,(G)$ as a function of M/N is greater than that of $H_d\,(G)$. This is due to the fact

that $L(G)$ has more nodes and edges than G, hence the loss of visibility makes the edge game significantly harder than the node game. There is one exception to the above remarks, namely the case $(M, N) = (1, 30)$; in this case both G and $L(G)$ are paths and $H_d(G)$ is essentially equal to $\overline{H}_d(G)$ (as can be seen by comparing Figures 6 and 10).

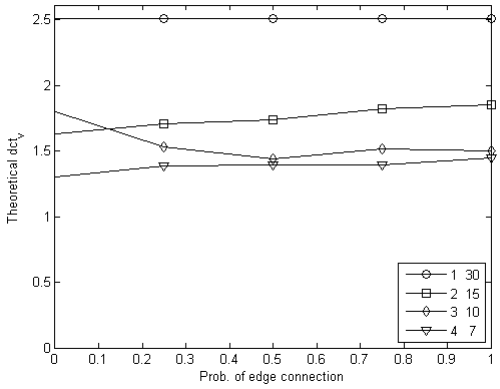

Figure 8: $\overline{dct}(G)$ curves for floorplans with n=30 or n=28 cells. Each curve corresponds to a fixed (M, N) pair. The horizontal axis corresponds to the edge insertion probability p_0.

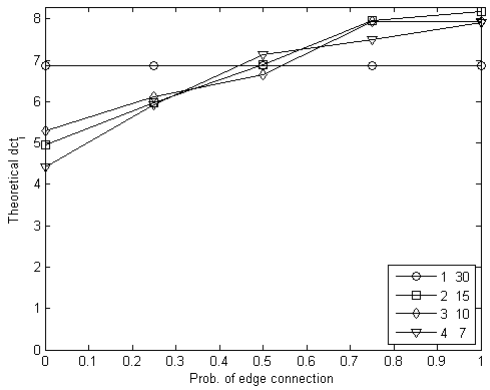

Figure 9: $\overline{dct}_i(G)$ curves for floorplans with n=30 or n=28 cells. Each curve corresponds to a fixed (M, N) pair. The horizontal axis corresponds to the edge insertion probability p_0.

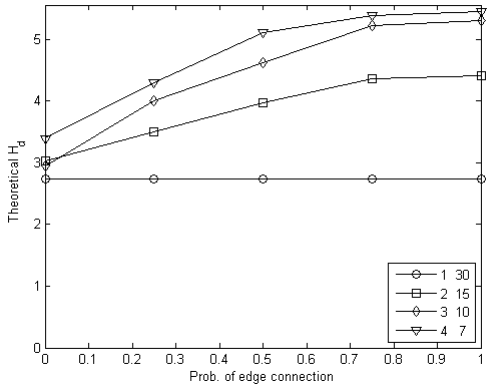

Figure 10: $\overline{H}_d(G)$ curves for floorplans with n=30 or n=28 cells. Each curve corresponds to a fixed (M, N) pair. The horizontal axis corresponds to the edge insertion probability p_0.

7 Conclusion

In this paper we have studied two versions of the cops and robber game: the one is played on the nodes of a graph and the other played on the edges. For each version, we studied four variants, obtained by changing the visibility and adversariality assumptions regarding the robber; hence we have a total of eight CR games. For each of these we have defined *rigorously* the corresponding optimal capture time, using game theoretic and probabilistic tools.

Then, for the node games we have introduced the adversarial cost of visibility $H(G) = \frac{ct_i(G)}{ct(G)}$ and the drunk cost of visibility $H_d(G) = \frac{dct_i(G)}{dct(G)}$. These ratios quantify the increase in difficulty of the CR game when the cop is no longer aware of the robber's position (this situation occurs often in mobile robotics).

We have defined analogous quantities ($\overline{H}(G) = \frac{\overline{ct}_i(G)}{\overline{ct}(G)}$, $\overline{H}_d(G) = \frac{\overline{dct}_i(G)}{\overline{dct}(G)}$) for the edge CR games.

We have studied analytically $H(G)$ and $H_d(G)$ and have established that both can get arbitrarily large. We have established similar results for $\overline{H}(G)$ and $\overline{H}_d(G)$. In addition, we have studied $H_d(G)$ and $\overline{H}_d(G)$ by numerical experiments which support both the game theoretic results of the current paper and the analytical computations of capture times presented in [23, 22].

References

[1] M. Adler, H. Racke, N. Sivadasan, C. Sohler, and B. Vocking, *Randomized pursuit-evasion in graphs*, Lecture Notes in Computer Science **2380** (2002), 901–912.

[2] M. Aigner and M. Fromme, *A game of cops and robbers*, Discrete Applied Mathematics **8** (1984), no. 1, 1–12.

[3] B. Alspach, *Searching and sweeping graphs: a brief survey*, Le Matematiche **59** (2006), no. 1–2, 5–37.

[4] D. Berwanger, *Graph games with perfect information*.

[5] A. Bonato and R. Nowakowski, *The game of cops and robbers on graphs*, AMS, 2011.

[6] A.Y Bonato and G. Macgillivray, *A general framework for discrete-time pursuit games*.

[7] T.H. Chung, G.A. Hollinger, and V. Isler, *Search and pursuit-evasion in mobile robotics*, Autonomous Robots **31** (2011), no. 4, 299–316.

[8] Robert Pallu de La Barrière, *Optimal control theory: a course in automatic control theory*, Dover Pubns, 1980.

[9] Dariusz Dereniowski, Danny Dyer, Ryan M Tifenbach, and Boting Yang, *Zero-visibility cops and robber game on a graph*, Frontiers in Algorithmics and Algorithmic Aspects in Information and Management, Springer, 2013, pp. 175–186.

[10] A. Dudek, P. Gordinowicz, and P. Pralat, *Cops and robbers playing on edges*, Journal of Combinatorics **5** (2013), no. 1, 131–153.

[11] JH Eaton and LA Zadeh, *Optimal pursuit strategies in discrete-state probabilistic systems*, Trans. ASME Ser. D, J. Basic Eng **84** (1962), 23–29.

[12] F.V. Fomin and D.M. Thilikos, *An annotated bibliography on guaranteed graph searching*, Theoretical Computer Science **399** (2008), no. 3, 236–245.

[13] B. Gerkey, S. Thrun, and G. Gordon, *Parallel stochastic hill-climbing with small teams*, Multi-Robot Systems. From Swarms to Intelligent Automata Volume III (2005), 65–77.

[14] G. Hahn and G. MacGillivray, *A note on k-cop, l-robber games on graphs*, Discrete mathematics **306** (2006), no. 19-20, 2492–2497.

[15] Milos Hauskrecht, *Value-function approximations for partially observable markov decision processes*, Journal of Artificial Intelligence Research **13** (2000), no. 1, 33–94.

[16] G. Hollinger, S. Singh, J. Djugash, and A. Kehagias, *Efficient multi-robot search for a moving target*, The International Journal of Robotics Research **28** (2009), no. 2, 201.

[17] G. Hollinger, S. Singh, and A. Kehagias, *Improving the efficiency of clearing with multi-agent teams*, The International Journal of Robotics Research **29** (2010), no. 8, 1088–1105.

[18] Ronald A Howard, *Dynamic probabilistic systems, volume ii: Semi-markov and decision processes*, John Wiley and Sons, New York, 1971.

[19] D. Hsu, W.S. Lee, and N. Rong, *A point-based POMDP planner for target tracking*, Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on, IEEE, 2008, pp. 2644–2650.

[20] V. Isler, S. Kannan, and S. Khanna, *Randomized pursuit-evasion with local visibility*, SIAM Journal on Discrete Mathematics **20** (2007), no. 1, 26–41.

[21] V. Isler and N. Karnad, *The role of information in the cop-robber game*, Theoretical Computer Science **399** (2008), no. 3, 179–190.

[22] Athanasios Kehagias, Dieter Mitsche, and Paweł Prałat, *Cops and invisible robbers: The cost of drunkenness*, Theoretical Computer Science **481** (2013), 100–120.

[23] Athanasios Kehagias and Paweł Prałat, *Some remarks on cops and drunk robbers*, Theoretical Computer Science **463** (2012), 133–147.

[24] H.W. Kuhn, *Extensive games*, Proceedings of the National Academy of Sciences of the United States of America **36** (1950), no. 10, 570.

[25] H. Kurniawati, D. Hsu, and W.S. Lee, *Sarsop: Efficient point-based POMDP planning by approximating optimally reachable belief spaces*, Proc. Robotics: Science and Systems, 2008.

[26] H. Lau, S. Huang, and G. Dissanayake, *Probabilistic search for a moving target in an indoor environment*, Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on, IEEE, 2006, pp. 3393–3398.

[27] M.L. Littman, A.R. Cassandra, and L.P. Kaelbling, *Efficient dynamic-programming updates in partially observable Markov decision processes*, Tech. Report CS-95-19, Comp. Science Dep., Brown University, 1996.

[28] René Mazala, *Infinite games*, Automata logics, and infinite games (2002), 197–204.

[29] G.E. Monahan, *A survey of partially observable Markov decision processes: Theory, models, and algorithms*, Management Science **28** (1982), no. 1, 1–16.

[30] R. Nowakowski and P. Winkler, *Vertex-to-vertex pursuit in a graph*, Discrete Mathematics **43** (1983), no. 2-3, 235–239.

[31] Martin J Osborne, *A course in game theory*, Cambridge, Mass.: MIT Press, 1994.

[32] J. Pineau and G. Gordon, *POMDP planning for robust robot control*, Robotics Research (2007), 69–82.

[33] M.L. Puterman, *Markov decision processes: Discrete stochastic dynamic programming*, John Wiley & Sons, Inc., 1994.

[34] T.E.S. Raghavan and J.A. Filar, *Algorithms for stochastic games - a survey*, Mathematical Methods of Operations Research **35** (1991), no. 6, 437–472.

[35] A. Sarmiento, R. Murrieta, and S.A. Hutchinson, *An efficient strategy for rapidly finding an object in a polygonal world*, Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on, vol. 2, IEEE, 2003, pp. 1153–1158.

[36] T. Smith and R. Simmons, *Heuristic search value iteration for POMDPs*, Proceedings of the 20th conference on Uncertainty in artificial intelligence, AUAI Press, 2004, pp. 520–527.

[37] M.T.J. Spaan and N. Vlassis, *Perseus: Randomized point-based value iteration for POMDPs*, Journal of artificial intelligence research **24** (2005), no. 1, 195–220.

[38] M. Vieira, R. Govindan, and G.S. Sukhatme, *Scalable and practical pursuit-evasion*, Robot Communication and Coordination, 2009. ROBOCOMM'09. Second International Conference on, IEEE, 2009, pp. 1–6.