The Role of Visibility in the Cops-Robber Game and Robotic Pursuit / Evasion

Athanasios Kehagias, Dieter Mitsche, and Paweł Prałat

***Abstract*— The cops-and-robber (CR) game has been used in mobile robotics as a discretized model of pursuit/evasion problems. The "classical" CR version is a *perfect information game*: the cop's (pursuer's) location is always known to the robber (evader) and vice versa. More relevant to robotics are versions where one (or both) player is invisible. In this paper we study the extent to which the CR game becomes harder (for the cop) when the robber is invisible. To this end we define the *cost of visibility* (*COV*) and study its existence and mathematical properties. We also compute COV analytically for particular graph families (paths, cycles, trees, grids). Finally, we compute COV numerically for several families of indoor environments; to perform this computation we introduce the heuristic algorithm *Pruned Cop Search* (PCS).**

I. Introduction

Pursuit / evasion (PE) problems have been the subject of extensive research in the last fifty years and much of this research has been connected to robotics [6]. An important subset of PE problems are *cops-and-robber* (CR) games [5] played on *graphs*. In the current paper, inspired by Isler and Karnad's recent work [15], we study the role of *information* in CR games.

By "information" we mean specifically the players' *location*. For example, we expect that when the cops know the robber's location (at every time t) they can do better than when the robber is "invisible". Our goal is to make the term "better" precise.

Pursuit / evasion and related problems (search, tracking, surveillance) often arise in mobile robotics; see the survey [6]. A graph can be used to represent topologically the environment (e.g., a building can be represented by a graph, with vertices corresponding to rooms and edges corresponding to doors). Hence the original problem is reduced to a *graph game* played between the pursuers and the evader; this approach has been utilized in several publications [3], [4], [30], [31]. If it is further assumed that the evader is not actively trying to avoid capture, the result is a *one-player* graph game; this model has been used quite often in mobile robotics [8], [11], [12], [20], [27] and especially (when assuming random evader movement) in publications such as [13], [19], [24], [28], [29], which utilize *partially observable Markov decision processes* (POMDP, [10], [21], [22]) .

Cops-and-robber variants form an important family of PE graph games. Reviews of the graph theoretic CR literature appear in [2], [5], [7]. In the "classical" CR variant [23] 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 cops) has received much less attention in the graph theoretic literature; among the few papers which treat this case we mention [14], [15], [16]; 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 [14], [15], [31]. But the invisible drunk robber CR variant is essentially identical to the POMDP model of PE which has been used in several of the previously mentioned robotic publications.

II. Preliminaries

Let $G = (V, E)$ be a fixed, finite, undirected, simple, connected graph with n nodes. One robber and K cops (with $K \geq 1$) move along the edges of G in discrete time steps $t \in \mathbb{N}_0 = \{0, 1, \ldots\}$. The robber's location is Y_t and the cops' locations are $X_t = (X_t^1, X_t^2, \ldots, X_t^K)$ ($t \in \mathbb{N}_0$ and $k \in \{1, 2, \ldots, K\}$). The game is played in *turns*; in every turn *first* the cops choose X_t and *then* the robber chooses Y_t. For all t and k, $\{X_{t-1}^k, X_t^k\} \in E$ or $X_{t-1}^k = X_t^k$; similarly, $\{Y_{t-1}, Y_t\} \in E$ or $Y_{t-1} = Y_t$ (i.e., only moves along edges of G are allowed). Once the robber is caught (which, in the adversarial version, can happen only after the cops' move; in the drunk version, it might also happen after the robber's move) he cannot move anymore. Given the complete sequences of cops and robber moves, the *capture time* is denoted by T and is defined as follows

$$T = \min\{t : \exists k \text{ such that } X_t^k = Y_t\}$$

with $T = \infty$ if no capture takes place. We assume the cops are *adversarial*: they play optimally to capture the robber in the smallest possible time. The robber can be in one of the two modes.

1) *Adversarial*: he wants to avoid capture for as long as possible and plays optimally toward this end.
2) *Drunk*: he performs a *random walk* on G such that, for all $\forall u, v \in V$ we have

$$\Pr(Y_0 = u) = \frac{1}{n} \qquad \text{and} \tag{1}$$

$$\Pr(Y_{t+1} = u | Y_t = v) = \begin{cases} \frac{1}{|N(v)|} & \text{iff } u \in N(v) \\ 0 & \text{otherwise} \end{cases}. \tag{2}$$

The cops' locations are always known to the robber but he can be either *visible* (his location is known to the cops) or *invisible* (his location is unknown). Hence we have four different CR variants, as detailed in Table I.

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

TABLE I
FOUR VARIANTS OF THE CR GAME.

In av-CR and ai-CR we assume that player C controls the cops and wants to minimize T; and player R controls the robber and wants to maximize T. Let R's *payoff* be T and C's payoff be $-T$; then av-CR and ai-CR are *two-player, zero-sum* games. In contrast, dv-CR and di-CR are one-player games (no strategy of R player is involved).

In the following sections we present several theorems; because of space limitations proofs are omitted; they can be found in [16], [17].

III. COP NUMBER AND CAPTURE TIMES

Our main goal in this section is to rigorously define the (expected) capture time for the various CR variants.

The av-CR Game. Since both cops and robber are visible and the players move sequentially, av-CR is a game of *perfect information* [18]. In such a game C loses nothing by limiting himself to *pure* (i.e., deterministic) strategies of the form $s_C(x,y)$: when the current cop / robber configuration is (x,y) and it is C's turn to play, $s_C(x,y)$ gives the next cop moves. Similarly R loses nothing by using pure strategies of the form $s_R(x,y)$. A pair (s_C, s_R) determines the entire course of the game; in other words, the capture time is a (deterministic) function $T(s_C, s_R)$. Since ai-CR is a game of perfect information, there exist *optimal* strategies $\widehat{s}_C$ and $\widehat{s}_R$ which satisfy

$$\min_{s_C}\max_{s_R} T(s_C, s_R) = T(\widehat{s}_C, \widehat{s}_R) = \max_{s_R}\min_{s_C} T(s_C, s_R).$$

$T(\widehat{s}_C, \widehat{s}_R)$ is the *value* of the game and also depends on K, the number of cops. This dependence appears explicitly in the notation of the following definition.

Definition 3.1: We denote by $ct(G,K)$ the optimal capture time $T(\widehat{s}_C, \widehat{s}_R)$ when CR is played on G by K adversarial, visible cops and one adversarial, visible robber.

Definition 3.2: The *cop number* of G is denoted by $c(G)$ and defined by $c(G) = \min\{K : ct(G,K) < \infty\}$.

Definition 3.3: The *visible capture time* of G is denoted by $ct(G)$ and defined by $ct(G) = ct(G, c(G))$.

It has been shown in [9] that, giveen $c(G)$, $ct(G)$ and the optimal search strategies $\widehat{s}_C$, $\widehat{s}_R$ can be computed in polynomial time using a game theoretic version of *value iteration* [25]. We have presented an implementation of this version (CAAR: Cops Against Adversarial Robber) in [17]. As shown in [9], the following holds.

Theorem 3.4: Given a graph G and a number $K \in \mathbb{N}$. If $K \geq c(G)$, the algorithm CAAR computes $ct(G,K)$ and the optimal search strategies $\widehat{s}_C$, $\widehat{s}_R$ in time $O\left(n^{2K+3}\right)$; conversely, if CAAR does not terminate then $K < c(G)$.

The dv-CR Game. Here the robber is visible and performs a random walk on G as indicated by (1)-(2). In the absence of cops, Y_t is a Markov chain on V, with transition probability matrix P. In the presence of one or more cops, $\{Y_t\}_{t=0}^{\infty}$ is a *Markov decision process* (MDP) [25] with state space $V \cup \{\lambda\}$ (λ is the *capture state*) and transition probability matrix $P(X_t)$ (i.e., X_t is the *control* variable, selected by C). It is shown in [17] how to obtain $P(x)$ from P. Hence T is a random variable depending on $X_0, X_1, \ldots$which C selects so as to minimize the expected capture time $E(T|X_0, X_1, \ldots)$. In fact (using standard results from [25]) C loses nothing by determining $X_0, X_1, \ldots$ through a strategy function $s_C(x,y)$; hence we can write $E(T|s_C)$. For every $K \geq 1$ and every s_C, $E(T|s_C)$ is well defined and finite; there exists an optimal strategy $\widehat{s}_C$ which minimizes $E(T|s_C)$. Hence we have the following.

Definition 3.5: We denote by $dct(G,K)$ the optimal expected capture time $E(T|\widehat{s}_C)$ when CR is played on G by K adversarial cops and one drunk visible robber. The *drunk visible capture time* of G is denoted by $dct(G)$ and defined by $dct(G) = dct(G, c(G))$.

For any given K, *value iteration* can be used to determine both $dct(G,K)$ and the optimal strategy $\widehat{s}_C(x,y)$; one implementation is our CADR (Cops Against Drunk Robber) algorithm [17], which has the following properties.

Theorem 3.6: Given a graph G and some $K \in \mathbb{N}$, CADR computes a sequence $\left\{\left(s_C^{(i)}, E\left(T|s_C^{(i)}\right)\right)\right\}_{i=1}^{\infty}$ such that: $\lim_{i\to\infty} s_C^{(i)} = \widehat{s}_c$ and $\lim_{i\to\infty} E\left(T|s_C^{(i)}\right) = E(T|\widehat{s}_C) = dct(G,K)$.

The ai-CR Game. In this game C cannot see R's moves: it is *not* a perfect information game. Both C and R must use *mixed* strategies σ_C, σ_R. A mixed strategy σ_C specifies, for every t, a conditional probability $\Pr(X_t|X_0, \ldots, X_{t-1})$ according to which C selects his t-th move; similarly σ_R specifies, for every t, a conditional probability $\Pr(Y_t|X_0, Y_0, \ldots, X_{t-1}, Y_{t-1}, X_t)$. A strategy pair (σ_R, σ_C), 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 gain (and C's loss), namely $E(T|\sigma_C, \sigma_R)$.

Denote the ai-CR game on a given G by Γ; since the game can last an infinite number of turns, it is not clear that Γ has a value. C can guarantee that he loses no more than $\sup_{\sigma_R}\inf_{\sigma_C} E(T|\sigma_C, \sigma_R)$ and R can guarantee that he gains no less than $\inf_{\sigma_C}\sup_{\sigma_R} E(T|\sigma_C, \sigma_R)$. We always have

$$\sup_{\sigma_R}\inf_{\sigma_C} E(T|\sigma_C, \sigma_R) \leq \inf_{\sigma_C}\sup_{\sigma_R} E(T|\sigma_C, \sigma_R); \quad (3)$$

by definition, the game will have a *value* if and only if equality is achieved in (3). We have shown in [16] that Γ *does* have a value for every graph G. To this end we have first proved that *invisibility does not increase the cop number*. More precisely, we have the following.

Theorem 3.7: On any graph G let $\overline{\sigma}_C$ denote the strategy by which $c(G)$ cops random-walk on G. Then $E(T|\overline{\sigma}_C, \sigma_R) < \infty$ for *every* robber strategy σ_R.

This is somewhat surprising because we expect that ai-CR is harder (from C's point of view) than av-CR. As we will see in Section III, the increased difficulty is reflected in the capture time, rather than in the number of cops.

Consider Γ_m, a "truncated" ai-CR game played exactly as Γ, but lasting m turns; R receives one unit of payoff for every turn in which the robber is not captured. The strategies σ_R and σ_C can be used in any truncated game Γ_m: C and R use them only until turn m. We denote the expected capture time for Γ_m by $E_m(T|\sigma_R, \sigma_C)$. Because Γ_m is a *finite*, two-person, zero-sum game, it has a value and optimal strategies. I.e., there exist strategies $\widehat{\sigma}_C^{(m)}$, $\widehat{\sigma}_R^{(m)}$ such that

$$\sup_{\sigma_R}\inf_{\sigma_C} E_m(T|\sigma_C, \sigma_R) = E_m\left(T|\widehat{\sigma}_C^{(m)}, \widehat{\sigma}_R^{(m)}\right) = \inf_{\sigma_C}\sup_{\sigma_R} E_m(T|\sigma_C, \sigma_R). \quad (4)$$

Hence the value of Γ_m is $val(\Gamma_m) = E_m\left(T|\widehat{\sigma}_C^{(m)}, \widehat{\sigma}_R^{(m)}\right)$, which is finite, because of Theorem 3.7. We also define

$$\underline{val}(\Gamma) = \sup_{\sigma_R}\inf_{\sigma_C} E(T|\sigma_R, \sigma_C),$$
$$\overline{val}(\Gamma) = \inf_{\sigma_C}\sup_{\sigma_R} E(T|\sigma_R, \sigma_C);$$

$\underline{val}(\Gamma) \leq \overline{val}(\Gamma)$ is always true; if equality holds then $val(\Gamma) = \underline{val}(\Gamma) = \overline{val}(\Gamma)$ is the *value* of the game Γ. The following theorem holds.

Theorem 3.8: Given any graph G and the corresponding CiR game Γ played with $c(G)$ cops, $val(\Gamma)$ exists and satisfies

$$val(\Gamma) = \lim_{m\to\infty} val(\Gamma_m).$$

Furthermore, there exists a strategy $\widehat{\sigma}_C$ such that

$$\sup_{\sigma_R} E(T|\sigma_R, \widehat{\sigma}_C) = val(\Gamma) \quad (5)$$

and for every $\varepsilon > 0$ there exists an m_ε and a strategy $\widehat{\sigma}_R^\varepsilon$ such that

$$\forall m \geq m_\varepsilon : val(\Gamma) - \varepsilon \leq \inf_{\sigma_C} E_m(T|\widehat{\sigma}_R^\varepsilon, \sigma_C). \quad (6)$$

Hence $val(\Gamma)$ can be approximated within any ε by using strategies $\widehat{\sigma}_R^\varepsilon$ and $\widehat{\sigma}_C$. Having established the existence of $val(\Gamma)$ we have the following.

Definition 3.9: Given a graph G, we define the *adversarial invisible capture time* of G to be

$$ct_i(G) = val(\Gamma)$$

where the game Γ is played with $c(G)$ cops.

The di-CR Game. Here Y_t is unobservable and controlled by X_t; hence Y_t is a POMDP. We denote this one-player game by $\overline{\Gamma}$ and the m-steps truncated version by $\overline{\Gamma}_m$. C must select a strategy σ_C which minimizes

$$E\left(T|\sigma_C, \overline{\Gamma}\right) = E\left(\sum_{t=0}^{\infty} \mathbf{1}(X_t \neq Y_t) |\sigma_C\right) + 1.$$

This is a typical *infinite horizon, undiscounted* POMDP problem [25]. $E_m(T|\sigma_C)$ and $E(T|\sigma_C)$ are well defined for every σ_C. We define $val\left(\overline{\Gamma}_m\right) = \inf_{\sigma_C\in\mathcal{S}_C} E_m(T|\sigma_C)$, $val\left(\overline{\Gamma}\right) = \inf_{\sigma_C\in\mathcal{S}_C} E(T|\sigma_C)$; the following holds.

Graph Family	Node Num.	$c(G)$	$H(G)$	$H_d(G)$
n node path	n	1	$2+o(1)$	$2+o(1)$
n node cycle	n	2	$2+o(1)$	$2+o(1)$
n node clique	n	1	$n-1$	$(1+o(1))\,n/2$
depth L d-ary tree	$n=\frac{d^L-1}{d-1}$	1	$\Theta(n)$	$\Theta\left(\frac{n}{\ln n}\right)$
$N\times N$ grid	$n=N^2$	2	$O\left(5^{\sqrt{n}}\sqrt{n}\right)$	$O\left(\sqrt{n}\cdot\log n\right)$

TABLE II
THE ADVERSARIAL AND DRUNK COST OF VISIBILITY FOR SEVERAL GRAPH FAMILIES.

Theorem 3.10: Given any graph G and the corresponding CiR game Γ, we have

$$val\left(\overline{\Gamma}\right) = \lim_{m\to\infty} val\left(\overline{\Gamma}_m\right).$$

Furthermore, there exists a strategy $\widehat{\sigma}_C$ such that

$$E\left(T|\widehat{\sigma}_C, \overline{\Gamma}\right) = val\left(\overline{\Gamma}\right).$$

Hence we can introduce the following definition.

Definition 3.11: Given a graph G, we define the *drunk invisible capture time* of G to be $dct_i(G) = val\left(\overline{\Gamma}\right)$.

Computation of Capture Times and Optimal Strategies. For graphs of relatively simple structure (e.g., paths, cycles, full trees, grids) capture times and optimal strategies can be computed exactly by analytical arguments. For more complicated graphs, certain *optimality equations* must be solved, which requires the use of computer algorithms such as CAAR and CADR. For details see [16], [17].

IV. THE COST OF VISIBILITY: RIGOROUS RESULTS

As already remarked, we expect that ai-CR is harder than av-CR (from C's point of view). We quantify this statement by introducing the *cost of visibility*.

Definition 4.1: For every G, the *adversarial cost of visibility* is $H(G) = \frac{ct_i(G)}{ct(G)}$.

Obviously we have $H(G) \geq 1$ (capturing the invisible robber is at least as hard as capturing the visible one). The following theorem shows that $H(G)$ can be arbitrarily large and, in fact, can approximate any real number $a \in [2, \infty)$.

Theorem 4.2: For every $a \in [2, \infty)$ there exists a family of graphs $\{G_n\}_{n=1}^{\infty}$ such that: (a) $\forall n : G_n$ has n nodes; (b) $\lim_{n\to\infty} H(G_n) = a$.

We define the *drunk cost of visibility* analogously.

Definition 4.3: The *drunk cost of visibility* is defined as $H_d(G) = \frac{dct_i(G)}{dct(G)}$.

$H_d(G)$, similarly to $H(G)$, can be arbitrarily large and can approximate any real number in $[1, \infty)$.

Theorem 4.4: For every $a \in [1, \infty)$ there exists a family of graphs $\{G_n\}_{n=1}^{\infty}$ such that: (a) $\forall n : G_n$ has n nodes; (b) $\lim_{n\to\infty} H_d(G_n) = a$.

Note that in Theorem 4.4 the range of attainable H_d values is $[1, \infty)$ while in Theorem 4.2 the range of attainable H values is $[2, \infty)$. We have not been able to find a graph G with two or more nodes and $H(G) \in (1, 2)$. We refer to this as a *COV gap*; we conjecture that there exists no graph G such that $|V(G)| \geq 2$ and $H(G) \in (1, 2)$. From previously obtained [16], [17] capture time results, we can determine analytically the (adversarial / drunk) COV for various graph families, as seen in Table II.

V. The Cost of Visibility: Experimental Results

We now present numerical computations of $H_d(G)$ (the drunk cost of visibility) for graphs G which are not amenable to analytical computation. Since $H_d(G) = \frac{dct_i(G)}{dct(G)}$, we use CADR to compute $dct(G)$ and PCS (an algorithm presented in the Appendix) to compute $dct_i(G)$[1]. We use graphs obtained from *indoor environments*, which we represent by their *floorplans*. In Fig. 1 we present a floorplan and its graph representation. The graph is obtained by decomposing the floorplan into convex *cells*, corresponding each cell to a node and connecting nodes by edges whenever the respective cells are connected by an open space.

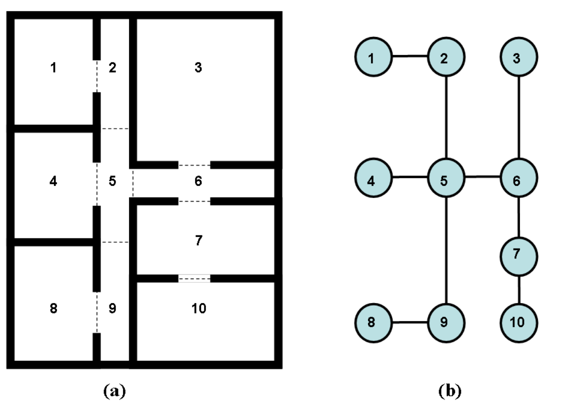

Fig. 1. Reducing a floorplan to a graph.

Setup. We have written a Matlab script which, given some parameters, generates random floorplans and their graphs. Every floorplan consists of a rectangle divided into orthogonal "rooms". If each room were connected to each four nearest neighbors we would get an $M \times N$ grid G'. However, we randomly generate a spanning tree T of G' and initially introduce doors only between rooms which are connected in T. Our final graph G is obtained from 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.

Experiment. We vary the ratio M/N. Namely we use the following pairs of (M, N) values: (1,30), (2,15), (3,10), (4,7), (5,6). All of these pairs give a total of 30 nodes (except that $M = 4$, $N = 7$ gives $n = 28$) and as M/N increases, we progress from a path to a nearly square grid. For each (M, N) pair we use three p_0 values: $\{0.0, 0.5, 1.0\}$; note the progression from a tree ($p_0 = 0.0$) to a full grid ($p_0 = 1.0$).

[1] 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 $H(G)$. 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 [26]).

For each triple (M, N, p_0) we generate 100 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. 2 we plot $dct(G)$ as a function of the ratio $c = M/N$; each curve corresponds to a different value of p_0. At the minimum value $c = \frac{1}{30}$ the graph is a path with capture time $dct(G) = \frac{N}{8}$ (recall that the problem involves *two* cops). At the maximum value $c = \frac{5}{6}$ the graph is a nearly square grid and capture time is close to $\frac{3M}{8}$ for the full grid (it appears that the p_0 value does not significantly influence capture time). Similar results can be observed in Fig. 3, in which $dct_i(G)$ is plotted. In Fig. 4 we plot the COV $H_d(G)$. We see that $H_d(G)$ is an

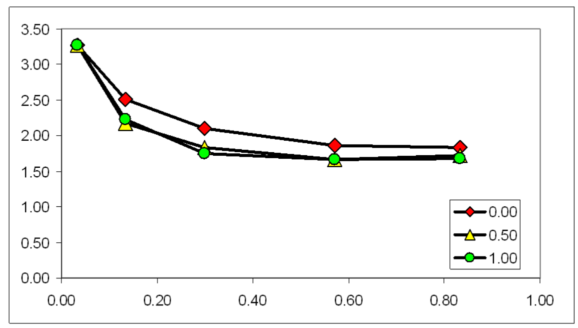

Fig. 2. $dct(G)$ curves for floorplans with $n = 30$ cells. Each curve corresponds to a fixed p_0 value. The horizontal axis corresponds to the ratio $c = M/N$ of horizontal to vertical dimension.

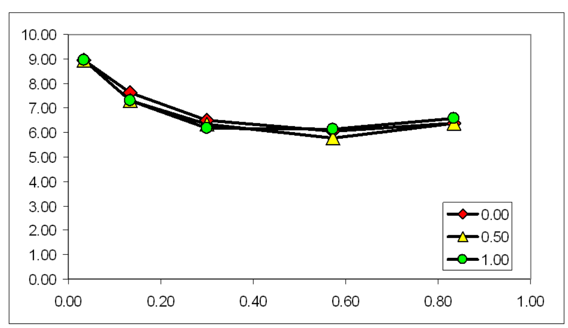

Fig. 3. $dct_i(G)$ curves for floorplans with $n = 30$ cells. Each curve corresponds to a fixed p_0 value. The horizontal axis corresponds to the ratio $c = M/N$ of horizontal to vertical dimension.

increasing function of $c = M/N$.

VI. Conclusion

In this paper we have studied four variants of the cops-and-robber game, obtained by changing the visibility and adversariality assumptions regarding the robber. For each

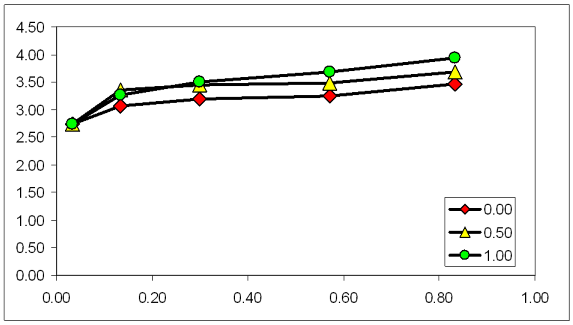

Fig. 4. $H_d(G)$ curves for floorplans with $n = 30$ cells. Each curve corresponds to a fixed p_0 value. The horizontal axis corresponds to the ratio $c = M/N$ of horizontal to vertical dimension.

variant we have *rigorously* defined the corresponding optimal capture time, using game theoretic and probabilistic tools. Then 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 studied $H_d(G)$, the drunk COV, analytically and computationally. On the analytical side we have shown that it can become arbitrarily large. In fact, $H_d(G)$ can approximate (by appropriate selection of G) any real number in $[1, \infty)$. This result emphasizes the importance of information about target location: the pursuit of a mobile target can become arbitrarily harder when the target's location is unknown. We have complemented our analytical study with numerical experiments on graphs obtained from artificially generated floorplans.

We have also studied $H(G)$, the adversarial COV, analytically and have established that it too can get arbitrarily large and can approximate any number in the interval $[2, \infty)$. We have not been able to find a graph for which $H(G)$ takes a value in $[1, 2]$. It appears then that a "COV gap" exists, which merits further study. It is possible that some G with $H(G) \in [1, 2]$ exists but it is hard to find; however we conjecture that this is not the case. More precisely, our conjecture is:

$$\forall G = (V, E) \text{ with } |V| > 1 \text{ we have } H(G) \geq 2.$$

We intend to further research this conjecture.

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

[3] A. Antoniades, H.J. Kim, and S. Sastry, *Pursuit-evasion strategies for teams of multiple agents with incomplete information*, Decision and Control, 2003. Proceedings. 42nd IEEE Conference on, vol. 1, IEEE, 2003, pp. 756–761.

[4] N. Basilico, N. Gatti, and F. Amigoni, *Leader-follower strategies for robotic patrolling in environments with arbitrary topologies*, Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 1, International Foundation for Autonomous Agents and Multiagent Systems, 2009, pp. 57–64.

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

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

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

[8] 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.

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

[10] M. Hauskrecht, *Value-function approximations for partially observable Markov decision processes*, Arxiv preprint arXiv:1106.0234 (2011).

[11] 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.

[12] 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.

[13] 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.

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

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

[16] A. Kehagias, D. Mitsche, and P. Prałat, *Cops and invisible robbers: the cost of drunkenness*, Arxiv preprint arXiv:1201.0946 (2012).

[17] A. Kehagias and P. Prałat, *Some remarks on cops and drunk robbers*, Arxiv preprint arXiv:1106.2414 (2011).

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

[19] 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.

[20] 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.

[21] 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.

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

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

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

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

[26] 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.

[27] 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.

[28] 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.

[29] 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.

[30] R. Vidal, O. Shakernia, H.J. Kim, D.H. Shim, and S. Sastry, *Probabilistic pursuit-evasion games: theory, implementation, and experi-*

mental evaluation, IEEE Transactions on Robotics and Automation **18** (2002), no. 5, 662–669.

[31] 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.

APPENDIX

In this appendix we briefly present *Pruned Cop Search* (PCS), an algorithm which computes $dct_i(G)$ heuristically. We give below the algorithm in pseudocode, denoting $E(T|X)$ by $U_C(X)$.

Algorithm 1 The Pruned Cop Search (PCS) Algorithm

Input: Graph $G = (V, E)$, initial cop position X_0, max. size of schedules list J_{max}, termination criterion ϵ
$t = 0$
$S.X = X_0$, $S.p = \Pr(Y_0|X_0)$, $S.U_C = 0$
$\mathbf{S} = \{S\}$
$\hat{U}_C^{old} = 0$
Continue=TRUE
while Continue **do**
  $\mathbf{S}' = \emptyset$
  for all $S \in \mathbf{S}$ **do**
    $X = S.X$, $p = S.p$, $U_C = S.U_C$
    $u = X_t$
    for all $v \in N^+(u)$ **do**
      $X' = X|v$
      $p' = pP(v)$
      $U'_C = \mathbf{Cost}(X', p', U_C)$
      $S'.X = X'$, $S'.p = p'$, $S'.U_C = U'_C$
      $\mathbf{S} = \mathbf{S} \cup \{S'\}$
    end for
  end for
  $\mathbf{S} = \mathbf{Prune}(\mathbf{S}', J_{max})$
  $[\hat{X}, \hat{U}_C] = \mathbf{Best}(\mathbf{S})$
  if $|\hat{U}_C - \hat{U}_C^{old}| < \epsilon$ **then**
    Continue=FALSE
  else
    $\hat{U}_C^{old} = \hat{U}_C$
    $t \leftarrow t + 1$
  end if
end while
Output: Best Schedule $\hat{X}$, Best Cost $\hat{U}_C$.

Here is a brief description of the algorithm rationale. PCS computes walks X which optimize $E(T|X)$ by a simple heuristic. A list of candidate search schedules is maintained; at the end of the $(t-1)$-st iteration each schedule in the list has length $t-1$; at the t-th iteration, each schedule is extended by one step and the value of the target function is approximated using the t-long search schedule. To avoid an exponential increase of the candidate search schedules, at the end of each iteration, the J_{max} best walks are retained and the remaining walks are removed.

The algorithm works because $E(T|X)$ can be approximated from a finite part of X, as explained below. We have

$$U_C(X) = E(T|X) = \sum_{t=0}^{\infty} t \cdot \Pr(T = t|X) = \sum_{t=0}^{\infty} \Pr(T > t|X). \quad (7)$$

X in the conditioning is the infinite walk X. However, for every t we have

$$\Pr(T > t|X) = 1 - \Pr(T \le t|X) = 1 - \Pr(T \le t|X_0, \dots, X_t).$$

Let us define

$$U_C^{(t)}(X_0 \dots X_t) = \sum_{\tau=0}^{t} (1 - \Pr(T \le \tau|X_0, \dots, X_\tau)) = \sum_{\tau=0}^{t} (1 - p_{n+1}(\tau)),$$

where $p_{n+1}(\tau)$ is the probability that the robber is in the capture state $n+1$ at time τ (the dependence on $X_0 X_1, ..., X_\tau$ is suppressed, for simplicity of notation). Then we have

$$U_C^{(t)}(X_0 \dots X_\tau) = U_C^{(t-1)}(X_0 \dots X_{t-1}) + (1 - p_{n+1}(t)). \quad (8)$$

Update (8) can be computed by the subroutine $\mathbf{Cost}\left(X^{(t)}, p, U_C^{(t-1)}\right)$ using only the previous cost $U_C^{(t-1)}(X_0 \dots X_{t-1})$ and the (already computed) probability vector $p(t)$. While $U_C^{(t)}(X_0 \dots X_t) \le U_C(X)$, we hope that (at least for the "good" walks) we have $\lim_{t\to\infty} U_C^{(t)}(X_0 \dots X_t) = U_C(X)$. This actually works well in practice.