Some remarks on cops and drunk robbers

Athanasios Kehagias [a], Paweł Prałat [b],*

[a] Department of Mathematics, Physics and Computer Sciences, Aristotle University of Thessaloniki, Thessaloniki GR54124, Greece
[b] Department of Mathematics, Ryerson University, Toronto, ON, Canada, M5B 2K3

ABSTRACT

The cops and robber game has been extensively studied under the assumption of optimal play by both the cops and the robber. In this paper we study the problem in which cops are chasing a *drunk* robber (that is, a robber who performs a random walk) on a graph. Our main goal is to characterize the "cost of drunkenness". Specifically, we study the ratio of expected capture times for the optimal version and the drunk robber one. We also examine the algorithmic side of the problem; that is, how to compute near-optimal search schedules for the cops. Finally, we present a preliminary investigation of the *invisible* robber game and point out differences between this game and *graph search*.

© 2012 Elsevier B.V. All rights reserved.

1. Introduction

The game of *Cops and Robber*, introduced independently by Nowakowski and Winkler [20] and Quilliot [26] almost thirty years ago, is played on a fixed undirected, simple, and finite graph G. There are two players, a team of k cops, where $k \geq 1$ is a fixed integer, and the *robber*. In the first round of the game, the cops occupy any set of k vertices and then the robber chooses a vertex to start from; in the following rounds, first the cops and then the robber move from vertex to vertex, following the edges of G. More than one cop is allowed to occupy a vertex, and the players may remain on their current positions. At every step of the game, both players know the positions of all cops and the robber. The cops win if they capture the robber; that is, if at least one of cops eventually occupies the same vertex as the robber; the robber wins if he can avoid being captured indefinitely. The players are *adversarial*; that is, they play optimally against each other. Since placing a cop on each vertex guarantees that the cops win, we may define the *cop number*, written $c(G)$, to be the minimum number of cops needed to win on G. The cop number was introduced by Aigner and Fromme in [1].

In this paper we study a new version of the game, in which the robber is *drunk*; that is, he performs a random walk on G. (In this paper, we consider the "standard" random walk: at every time step the robber will move equiprobably to a neighboring vertex. Another natural approach would be to examine the "lazy" random walk, in which the robber may also stay in place. Our results can be easily modified to the new setting.) The cops are assumed to follow a strategy which is optimal with respect to the robber's random behavior. This version was proposed by D. Thilikos during the 4th Workshop on GRAph Searching, Theory and Applications (GRASTA 2011, held in Schloss Dagstuhl) and he specifically asked the following question: "what is the *cost of drunkenness*?" In other words, how much faster than the adversarial robber is the drunk one captured? We try to answer various versions of this question. In addition, we study some algorithmic questions; for example, how to compute the expected capture time for an optimal strategy of cops.

There is a large bibliography on pursuit games on graphs. The most important open problem in this area is Meyniel's conjecture (communicated by Frankl [10]). It states that $c(n) = O(\sqrt{n})$, where $c(n)$ is the maximum of $c(G)$ over all n-vertex connected graphs. Recent years have witnessed significant interest in the study of random graphs from that perspective

---

* Corresponding author. Tel.: +1 416 979 5000x7421.
E-mail addresses: kehagiat@auth.gr (A. Kehagias), pralat@ryerson.ca (P. Prałat).

[4,6,17,22] confirming that, in fact, Meyniel's conjecture holds asymptotically almost surely for binomial random graphs [24] as well as random *d*-regular graphs [23]. The reader interested in cops and robbers can start by perusing the surveys [2,9,11] and the recent book [5]. To the best of our knowledge, the problem of a drunk robber has not been previously studied in the cops and robbers literature. However there is a strong connection to the *Markov Decision Processes* (MDP) literature; we will comment on this connection (and use it) in Section 5. The reader can refer to [15,25,27] for MDP surveys.

While the emphasis of the current paper is on cops chasing the visible robber, we also touch briefly the case of an *invisible* robber, both adversarial and drunk. Not much has been written on this problem, but a related problem which has been extensively studied is the *Graph Search* problem, where a team of searchers try to locate in a graph an *invisible* fugitive, who is also assumed to be *arbitrarily fast* and *omniscient* (he always knows the searchers' locations as well as their strategy). A recent comprehensive review of graph search appears in [9]. We emphasize that the graph search problem is similar but *not* identical to cops chasing an invisible robber.

The paper is structured as follows. In Section 2 we present definitions and our notation; the formulation is, naturally, probabilistic. In particular, we define the cost of drunkenness to be the ratio of the capture time for the adversarial robber and the *expected* capture time for the drunk robber. We also present a number of lemmas which we will repeatedly use in the following sections. In Section 3 we obtain bounds on the cost of drunkenness for various special families of graphs; for example, paths, cycles, grids, and complete *d*-ary trees. In Section 4 we look at the problem more generally and show that, for any $c \in [1, \infty)$, there is a graph for which the cost of drunkenness is arbitrarily close to c. In Section 5 we connect the cops and drunk robber problem to *Markov Decision Processes* (MDP); that is, Markov chains with a *control input* which can modify the transition probabilities. MDP's provide a natural language for the problem; in particular they are useful in the computation of optimal cop strategies; that is, strategies which minimize the expected robber capture time. We then use the MDP machinery to present algorithms which compute the optimal cop strategy for a given graph and a drunk robber. In Section 6 we give a brief, preliminary discussion of the cost of drunkenness for an *invisible* robber. Finally, in Section 7 we list possible future research directions.

2. Preliminaries

2.1. Definitions

Let $G = (V, E)$ be a fixed undirected, simple, and finite graph. Since the game played on a disconnected graph can be analyzed by investigating each component separately, we assume that G is connected. We will use the following notation and assumptions.

(i) Recall that the *cop number* $c(G)$ is the minimum number of cops needed to win on G.
(ii) There are k cops (for the time being we assume $k \geq c(G)$ but this assumption will be relaxed in later sections).
(iii) X_t^i denotes the position of the *i*-th cop at time t ($i \in \{1, 2, \ldots, k\}$, $t \in \{0, 1, 2, \ldots\}$); $X_t = (X_t^1, \ldots, X_t^k)$ denotes the vector of all cop positions at time t; $\mathbf{X} = (X_0, X_1, X_2, \ldots)$ denotes the positions of all cops during the game ($\mathbf{X}$ may have finite or infinite length).
(iv) Y_t denotes the position of the robber at time t and $\mathbf{Y} = (Y_0, Y_1, Y_2, \ldots)$ the positions of the robber during the game. (Let us note that there is a correlation between $\mathbf{X}$ and $\mathbf{Y}$; that is, players adjust their strategies observing moves of the opponent.)
(v) The moving sequence is as follows: first the cops choose initial positions $X_0 \in V$, then the robber chooses $Y_0 \in V$. For $t \in \{1, 2, \ldots\}$ first the cops choose X_t and then the robber chooses Y_t. Players use edges of the graph G to move from one vertex to another. In other words, for $i \in \{1, 2, \ldots, k\}$ and $t \in \{0, 1, 2, \ldots\}$, either $\{X_t^i, X_{t+1}^i\} \in E$ or $X_t^i = X_{t+1}^i$. Similarly, for $t \in \{0, 1, 2, \ldots\}$, either $\{Y_t, Y_{t+1}\} \in E$ or $Y_t = Y_{t+1}$ (in the drunk robber case we do not have $Y_t = Y_{t+1}$, see below).
(vi) Given a play X, Y, the *capture time* is denoted by T and defined as follows

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

that is, it is the first time a cop is located at the same vertex as the robber. In general, $X_t^i = Y_t$ can happen either after the cops move or, if the robber does not play optimally, after the robber moves; in the former case we have $X_t^i = Y_{t-1}$ and (assuming the robber cannot move any longer) also $Y_{t-1} = Y_t$; hence the definition of T is correct. Finally, note that $T < \infty$, since $k \geq c(G)$ and $c(G)$ cops can capture the adversarial robber (and so, of course, the drunk one too).

Assuming for the moment adversarial cops *and* robber, and given initial cop positions $x \in V^k$ and robber position $y \in V$, we let $\text{ct}_{x,y}(G, k) = T$. The *k-capture time* is defined as follows:

$$\text{ct}(G, k) = \min_{x \in V^k} \max_{y \in V} \text{ct}_{x,y}(G, k).$$

In other words, we allow our perfect players to choose their initial positions in order to achieve the best outcome. Finally, when $k = c(G)$ we simply write $\text{ct}(G)$ instead of $\text{ct}(G, c(G))$, and call it the *capture time* instead of the $c(G)$-capture time. Let us stress one more time that the above quantities are defined under the assumption of *optimal play by both players*.

Next let us assume that the cops are adversarial but that the robber is *drunk*. More specifically, we assume that the robber performs a random walk on G. Given that he is at vertex $v \in V$ at time t, he moves to $u \in N(v)$ at time $(t+1)$ with probability equal to $1/|N(v)|$. Note that we do *not* include v in $N(v)$; that is, we consider open, not closed, neighborhoods. Moreover, the robber probability distribution does not depend on current position of cops; in particular, it can happen that the robber moves to a vertex occupied by a cop (something the adversarial robber would never do).

Under the above assumptions, the drunk robber game is actually a one-player game and, for a given initial configuration and cops strategy, the capture time T is a random variable. For any $x \in V^k$ and $y \in V$, let

$$\text{dct}_{x,y}(G, k) = \mathbb{E}(T \mid X_0 = x, Y_0 = y, k \text{ cops are used optimally});$$

in other words, it is the expected capture time given initial cops and robber configurations x, y and assuming optimal play by the k cops.

Since the robber is drunk, we cannot expect him to choose the most suitable vertex to start with—instead, he chooses an initial vertex uniformly at random. Cops are, of course, aware of this and so they try to choose an initial configuration so that the expected length of the game is as small as possible. Hence, we define the expected k-capture time as follows:

$$\text{dct}(G, k) = \min_{x \in V^k} \sum_{y \in V} \frac{\text{dct}_{x,y}(G, k)}{|V|}.$$

As before, we let $\text{dct}(G) = \text{dct}(G, c(G))$. We define the *cost of drunkenness* as follows

$$F(G) = \frac{\text{ct}(G)}{\text{dct}(G)}$$

and we obviously have $F(G) \geq 1$.

While we concentrate on the case $k = c(G)$, it is also natural to consider expected capture time $\text{dct}(G, k)$ for $k \neq c(G)$. The next theorem shows that this is well defined for any $k \geq 1$ (in particular, even for $k < c(G)$).

Theorem 2.1. $\text{dct}(G, k) < \infty$ for any connected graph G and $k \geq 1$.

Proof. Let $G = (V, E)$ be any connected graph, $D = D(G)$ be the diameter of G, and $\Delta = \Delta(G)$ be the maximum degree of G. Fix any vertex $v \in V$, place k cops on v, and let $X_t^i = v$ for all i and t (that is, cops never move; this is clearly a suboptimal strategy). For a given vertex $y \in V$ occupied by the drunk robber, the probability that he uses a shortest path from y to v to move straight to v is at least $(1/\Delta)^D$. This implies that, regardless of the current position of the robber at time t, the probability that he will be caught after at most D further rounds is at least $\varepsilon = (1/\Delta)^D$. Moreover, corresponding events for times $t + iD, i \in \mathbb{N} \cup \{0\}$ are mutually independent. Thus, we get immediately that

$$\mathbb{E}T = \sum_{t \geq 0} \mathbb{P}(T > t) \leq \sum_{t \geq 0} \mathbb{P}\left(T > \left\lfloor \frac{t}{D} \right\rfloor D\right)$$

$$= \sum_{i \geq 0} D \cdot \mathbb{P}(T > iD) \leq D \sum_{i \geq 0} (1 - \varepsilon)^i = \frac{D}{\varepsilon} = D\Delta^D < \infty,$$

and we are done. $\square$

Let us remark that sharper bounds can be obtained for the capture time of a drunk robber, even in the case that the cops are also drunk; for example see [7]. However, Theorem 2.1 will be sufficient for our needs.

2.2. Some useful lemmas

We will be using the following version of a well-known Chernoff bound many times so let us state it explicitly.

Lemma 2.2 ([14]). *Let X be a random variable that can be expressed as a sum $X = \sum_{i=1}^n X_i$ of independent random indicator variables, where $X_i \in \text{Be}(p_i)$ with (possibly) different $p_i = \mathbb{P}(X_i = 1) = \mathbb{E}X_i$. Then the following holds for $t \geq 0$:*

$$\mathbb{P}(X \geq \mathbb{E}X + t) \leq \exp\left(-\frac{t^2}{2(\mathbb{E}X + t/3)}\right),$$

$$\mathbb{P}(X \leq \mathbb{E}X - t) \leq \exp\left(-\frac{t^2}{2\mathbb{E}X}\right).$$

In particular, if $\varepsilon \leq 3/2$, then

$$\mathbb{P}(|X - \mathbb{E}X| \geq \varepsilon\mathbb{E}X) \leq 2\exp\left(-\frac{\varepsilon^2\mathbb{E}X}{3}\right).$$

Let us now consider the following (simple) random walk on $\mathbb{Z}$. Understanding the behavior of this Markov chain will be important in investigating simple families of graphs later. Let $X_0 = 0$, and for a given $t \geq 0$, let

$$X_{t+1} = \begin{cases} X_t + 1 & \text{with probability } 1/2 \\ X_t - 1 & \text{otherwise.} \end{cases}$$

It is known that with high probability, random variable X_t stays relatively close to zero. We make this precise below using the Chernoff bound.

Lemma 2.3. *Let $n \in \mathbb{N}$ and $c \in (2, \infty)$. For a simple random walk (X_t) on $\mathbb{Z}$ with $X_0 = 0$ we have that $|X_t| \leq c\sqrt{n \log n}$ for every $t \in \{0, 1, \ldots, n\}$ with probability at least $1 - 2n^{1-c^2/4}$.*

Proof. Fix $n \in \mathbb{N}$ and $c \in (2, \infty)$. Let us perform n steps of a simple random walk on $\mathbb{Z}$ starting with $X_0 = 0$. Let Y_t ($1 \leq t \leq n$) denote the number of times the process goes 'up' until time t. It is clear that $\mathbb{E}Y_t = t/2$ and

$$X_t = Y_t - (t - Y_t) = 2(Y_t - t/2).$$

For a given t, it follows from the Chernoff bound (Lemma 2.2) that

$$\begin{aligned}
\mathbb{P}\left(X_t < -c\sqrt{n \log n}\right) &= \mathbb{P}\left(Y_t \leq \frac{t}{2} - \frac{c}{2}\sqrt{n \log n}\right) \\
&\leq \exp\left(-\frac{(c\sqrt{n \log n}/2)^2}{2(t/2)}\right) \\
&\leq \exp\left(-\frac{c^2}{4}\log n\right) = n^{-c^2/4}.
\end{aligned}$$

A symmetric argument can be used to get that $X_t > c\sqrt{n \log n}$ with probability at most $n^{-c^2/4}$. Finally, from a union bound we get that the probability that there exists t ($1 \leq t \leq n$) with $|X_t| > c\sqrt{n \log n}$ is at most $n \cdot 2n^{-c^2/4} = 2n^{1-c^2/4}$. $\quad\square$

3. Bounds on the Cost of Drunkenness

In this section we place upper and lower bounds on the cost of drunkenness $F(G)$ when k cops are available. We emphasize the case $k = c(G)$ but also consider values of $k \neq c(G)$. We start with simple graphs (namely: paths, cycles, trees, and grids) in order to prepare for slightly more complicated families in the next section.

3.1. Paths and a Suboptimal Strategy

In this subsection we play the game on P_n, a path on n vertices ($V(P_n) = \{0, 1, \ldots, n-1\}$, $E(P_n) = \{\{i-1, i\} : i \in \{1, 2, \ldots, n-1\}\}$). Clearly, $c(P_n) = 1$; that is, one cop can catch the adversarial robber. Since the drunk robber is easier to catch than the adversarial one, let us study the drunk robber playing against a single cop.

In this subsection we will compute the expected capture time using a *suboptimal* strategy, namely starting the cop at $X_0 = 0$ and moving him to the other end until he reaches $n - 1$ (or until capture takes place). It is clear that this strategy achieves capture; furthermore (as will become apparent in the following sections) many optimal strategies can be analyzed using this suboptimal one.

Let $Z_t = Y_t - X_t$ be the distance between players at time t. If the drunk robber starts at vertex $k \in \{0, 1, \ldots, n-1\}$, we have $Z_0 = Y_0 = k$. (In order to simplify the argument, we allow players to "pass each other" which is never the case in the real game; that is, Z_t can be negative.) We can redefine the capture time as

$$T_n = T_n(k) = \min\{t : Z_t \leq 0\}.$$

Now, it is not so difficult to see the behavior of the sequence $(Z_t)_{t \geq 0}$. Note that at time t, the maximum distance between players is $n - 1 - t$, which implies that the robber will be caught in at most $n - 1$ steps. We have the following Markov chain to investigate: for $t \in \{0, 1, \ldots, n-2\}$, if $Z_t < n - 1 - t$, then

$$Z_{t+1} = \begin{cases} Z_t - 2 & \text{with probability } 1/2 \text{ (the robber goes toward the cop)} \\ Z_t & \text{with probability } 1/2 \text{ (the robber goes away from the cop).} \end{cases}$$

If $Z_t = n - 1 - t$ (that is, the robber occupies the end of the path), then $Z_{t+1} = Z_t - 2$ (deterministically).

Consider another Markov chain Z'_t, which has the following simple behavior: $Z'_0 = k$ and for every $t \geq 0$, $Z'_{t+1} = Z'_t - 2$ with probability $1/2$; otherwise $Z'_{t+1} = Z'_t$. Define $T' = \min\{t : Z_t \leq 0\}$. In other words, we will be chasing the robber on the infinite ray R ($V(R) = \mathbb{N} \cup \{0\}$, $E(R) = \{\{i-1, i\} : i \in \mathbb{N}\}$), which is slightly more difficult for the cop. Hence, it is easy to prove that $\mathbb{E}(T_n \mid Z_0 = k) \leq \mathbb{E}(T' \mid Z'_0 = k)$. Moreover, it is also easy (using a recursive argument) to show that $\mathbb{E}(T' \mid Z'_0 = k) = k$, and so $\mathbb{E}(T_n(k)) \leq k$. Now we are ready to show the following.

Theorem 3.1. *Consider that the cop starts on one end of the path P_n and moves toward the other end. Let T_n be the capture time, provided that the robber is drunk. Then,*

$$\frac{n}{2}\left(1 - O\left(\frac{\log n}{n}\right)\right) \leq \mathbb{E}T_n \leq \frac{n-1}{2}.$$

Before we move to the proof of this theorem let us mention that, in fact, with a slightly more sophisticated argument, it is possible to show that $\mathbb{E}T_n = n/2 - O(1)$.

Proof. Let $n \in \mathbb{N}$ and fix any $c > 2$. The robber starts his walk on a vertex $k \in \{0, 1, \ldots, n-1\}$. Let us note that he is captured after at most $n - 1$ steps of the process (deterministically); that is, $T_n(k) \leq n - 1$. As we already mentioned $\mathbb{E}T_n(k) \leq k$. Since the starting vertex for the robber is chosen uniformly at random, we get that $\mathbb{E}T_n \leq \sum_{k=0}^{n-1} k/n = (n-1)/2$, so it remains to investigate a lower bound.

Suppose first that $k \leq (n - 1) - c\sqrt{n \log n}$. It follows from Lemma 2.3 that the robber reaches the other end of the path with probability at most $2n^{1-c^2/4}$. If this is the case, we apply a trivial lower bound for $T_n(k)$, namely, $T_n(k) \geq 0$; otherwise we get that the (conditional) expectation for $T_n(k)$ is equal to k. Hence, $\mathbb{E}T_n(k) \geq k(1 - 2n^{1-c^2/4})$. Suppose now that $k > (n - 1) - c\sqrt{n \log n}$. Using Lemma 2.3 one more time, we get that with probability at least $1 - 2n^{1-c^2/4}$ the robber is not caught before time $k - c\sqrt{n \log n}$.

Since the starting vertex for the robber is chosen uniformly at random, we get that

$$\begin{aligned}
\mathbb{E}T_n &\geq \frac{1}{n}\sum_{k=0}^{n-1}\mathbb{E}T_n(k) \\
&\geq \frac{1}{n}\left(\sum_{k=0}^{n-1-c\sqrt{n \log n}} k + \sum_{k=n-c\sqrt{n \log n}}^{n-1}(k - c\sqrt{n \log n})\right)(1 - 2n^{1-c^2/4}) \\
&\geq \left(\frac{n-1}{2} - c^2 \log n\right)(1 - 2n^{1-c^2/4}).
\end{aligned}$$

For a given n, the parameter c can be adjusted for the best outcome. To get an asymptotic behavior, we can use, say, $c = 3$ to get that

$$\mathbb{E}T_n \geq \frac{n}{2}\left(1 - O\left(\frac{\log n}{n}\right)\right),$$

and the proof is complete. $\square$

The proof of the theorem actually gives us more. We get that with probability tending to 1 as $n \to \infty$, for all starting points for the robber ($k \in \{0, 1, \ldots, n - 1\}$), the cop needs $k + O(\sqrt{n \log n})$ moves to catch the robber.

3.2. Paths

We continue studying a visible robber on P_n but we now apply the optimal capture strategy (it is optimal for both adversarial and drunk robber). If n is odd, we start by placing a cop on vertex $(n - 1)/2$; if n is even we have two optimal strategies, the cop can start on $n/2$ or $n/2 - 1$. In any case, after selecting an initial vertex the strategy is the same: the cop keeps moving toward the robber. Except for initial placement, this is the strategy examined in the previous subsection and we have $\mathrm{ct}(P_n) = \lfloor n/2 \rfloor$. We easily get the following result.

Theorem 3.2.

$$\frac{n}{4}\left(1 - O\left(\frac{\log n}{n}\right)\right) \leq \mathrm{dct}(P_n) \leq \frac{n}{4}.$$

In particular, $\mathrm{dct}(P_n) = (1 + o(1))n/4$ and the cost of drunkenness is

$$F(P_n) = \frac{\mathrm{ct}(P_n)}{\mathrm{dct}(P_n)} = 2 + o(1).$$

Proof. As we already mentioned, after the robber selects his initial vertex to start from, the game is played essentially on a path of length at most $\lfloor n/2 \rfloor + 1$. From Theorem 3.1, we get immediately that

$$\mathrm{dct}(P_n) \leq \mathbb{E}T_{\lfloor n/2 \rfloor + 1} \leq n/4.$$

For a lower bound, we notice that the length of each subpath is at least $\lfloor n/2 \rfloor$. By Theorem 3.1,

$$\mathrm{dct}(P_n) \geq \mathbb{E}T_{\lfloor n/2 \rfloor} \geq \frac{n}{4}\left(1 - O\left(\frac{\log n}{n}\right)\right),$$

and the proof is complete. $\square$

In the general case when $k \in \mathbb{N}$ cops are available, we need to 'slice' a path into k shorter paths and place a cop on their centers. We get that $\mathrm{dct}(P_n, k) = (1 + o(1))n/(4k)$.

3.3. Cycles

Let us play the game on a cycle C_n for $n \geq 4$ $(V(C_n) = \{1, 2, \ldots, n\}, E(C_n) = \{\{i, i+1\} : i \in \{1, 2, \ldots, n-1\}\} \cup \{\{1, n\}\})$. It is not difficult to see that $c(C_n) = 2$; we use two cops to chase the robber. They start by occupying two vertices at the distance $\lfloor (n + 1)/2 \rfloor$, the maximum possible distance on the cycle. When the robber selects his vertex to start with, they move toward him and capture occurs at time $\mathrm{ct}(C_n) = \lfloor (n + 1)/4 \rfloor$. The same strategy is used when the robber is drunk.

As for paths, one can introduce a random variable Z_t to measure the distance between the robber and cops at time t. The problem (almost) reduces to the problem on a path. We mention briefly the difference below but the formal proof is omitted. If n is odd, then Z_t has exactly the same behavior as before. However, $Z_0 = \lfloor (n + 1)/2 \rfloor$ with probability two times smaller than any other legal starting value (note that a uniform distribution on $V(C_n)$ is used but there is just one vertex at the distance $\lfloor (n + 1)/2 \rfloor$). If n is even, then we get a uniform distribution for starting values but the transition from Z_t to Z_{t+1} is slightly different, namely, there is a chance for Z_t to stay at the same value, provided that the robber occupies the vertex which is at the maximum distance from cops. In any case, it is straightforward to show that both upper and lower bounds still hold, so we get the following.

Theorem 3.3.

$$\frac{n}{8} \left(1 - O\left(\frac{\log n}{n}\right)\right) \leq \mathrm{dct}(C_n) \leq \frac{n+1}{8}.$$

In particular, $\mathrm{dct}(C_n) = (1 + o(1))n/8$ and the cost of drunkenness is

$$F(C_n) = \frac{\mathrm{ct}(C_n)}{\mathrm{dct}(C_n)} = 2 + o(1).$$

In the general case when $k \in \mathbb{N}$ cops are available, we spread them as evenly as possible. We get that $\mathrm{dct}(C_n, k) = (1 + o(1))n/(4k)$.

3.4. Trees

All families of graphs we discussed so far have a very nice property, namely, it is clear what the optimal strategy for the cops is. Once players fix their initial positions (that is, X_0 and Y_0), cops must move toward the robber in order to decrease the expected capture time. As we mentioned before, it is natural to measure the distance Z_t between players at time t; Z_t decreases by 2 if the robber makes a bad move or is occupying a leaf; otherwise the distance remains the same. This applies to the family of trees as well (note that $c(T) = 1$ for any tree T). However, this time it is not clear which vertex should be used for the cop to start with in order to optimize the expected capture time. For this family, the random variable Z_t decreases with probability $1/\deg(v)$, provided that the robber occupies vertex v, and the behavior of the sequence $(Z_t)_{t \geq 0}$ highly depends not only on the degree distribution but on the structure of a tree as well. It is non-trivial to estimate the cost of drunkenness for a particular tree without performing extensive calculations for every vertex as a starting point (these calculations can be performed by computer, using the algorithms of Section 5.2). However, some sub-families of trees are still relatively easy to deal with.

Let us consider d regular, rooted tree $T(d, k)$ of depth k, with $d \geq 2$ (in the case $d = 1$, $T(d, k)$ is a path and $F(T(d, k))$ is given by Theorem 3.2). The root vertex on the level 0 has d neighbors (children), vertices on levels 1 to $k-1$ have degree $d+1$ (one parent and d children), leaves on the level k have degree 1 (just one parent). There are d^i vertices on level i for a total of $(d^{k+1} - 1)/(d - 1)$ vertices. Due to the symmetry, the cop must start the game on the root. Since the drunk robber prefers to move toward leaves, it is natural to expect that his behavior is similar to the one of the adversarial robber. Moreover, almost all vertices are located on levels $k - o(k)$ so the robber almost always starts on these vertices, which is clearly a good move. We show that the cost of drunkenness is the best possible; that is, $\mathrm{dct}(T(d, k))$ is tending to $\mathrm{ct}(T(d, k)) = k$ as $k \to \infty$.

Theorem 3.4. *For $T(d, k)$ with $d \geq 2$ we have*

$$k - O(\sqrt{k \log k}) \leq \mathrm{dct}(T(d, k)) \leq k.$$

In particular, $\mathrm{dct}(T(d, k)) = (1 + o(1))k$ and the cost of drunkenness is

$$F(T(d, k)) = \frac{\mathrm{ct}(T(d, k))}{\mathrm{dct}(T(d, k))} = 1 + o(1).$$

Proof. Suppose that the drunk robber starts on level $i \geq k - \sqrt{k \log k}$. It follows from Lemma 2.2 that with probability $1 - O(k^{-1})$ he will be caught on level $k - O(\sqrt{k \log k})$. (In fact, it is also true for $i \geq k/d$, since the robber moves toward

leaves with higher rate, namely, with probability $(d-1)/d$. However, an error following from this part is negligible compared to the other error, so we stay with this obvious bound for i.) Therefore,

$$\mathrm{dct}(T(d, k)) \geq \sum_{i=k-\sqrt{k\log k}}^{k} \frac{d^i}{(d^{k+1}-1)/(d-1)}(k - O(\sqrt{k\log k}))(1 - O(k^{-1}))$$
$$= (1 - O(d^{-\sqrt{k\log k}}))(k - O(\sqrt{k\log k}))(1 - O(k^{-1}))$$
$$= k - O(\sqrt{k\log k}),$$

which finishes the proof. □

3.5. Grids

The *Cartesian product* of two graphs G and H is a graph with vertex set $V(G) \times V(H)$ and with the vertices (u_1, v_1) and (u_2, v_2) adjacent if either $u_1 = u_2$ and v_1, v_2 are adjacent in H, or $v_1 = v_2$ and u_1, u_2 are adjacent in G. We denote the Cartesian product of G and H by $G\square H$. In this subsection, we will study a square grid $P_n\square P_n$.

It is known that for any two trees T_1, T_2, each of which has at least two vertices, we have $c(T_1\square T_2) = 2$ [19]. The capture time of the Cartesian product of trees was recently studied in [18]. It was shown that for any two trees T_1, T_2 we have

$$\mathrm{ct}(T_1\square T_2) = \left\lfloor \frac{D(T_1\square T_2)}{2} \right\rfloor = \left\lfloor \frac{D(T_1) + D(T_2)}{2} \right\rfloor,$$

where $D = D(G)$ is the diameter of G. In particular, for a square grid we have that $\mathrm{ct}(P_n\square P_n) = n - 1$.

We will show that the cost of drunkenness for a grid is asymptotic to $8/3$.

Theorem 3.5.

$$\mathrm{dct}(P_n\square P_n) = (1 + o(1))\frac{3}{8}n,$$

and the cost of drunkenness is $F(P_n\square P_n) = 8/3 + o(1)$.

Proof. Suppose that the drunk robber occupies an internal vertex (u, v). The decision where to go from there can be made in the following way: toss a coin to decide whether to modify the first coordinate (u) or the second one (v); independently, another coin is tossed to decide whether we increase or decrease the value. Hence the robber will move with probability $1/4$ to one of the four neighbors of (u, v). Note that, if we restrict ourselves to look at one dimension only (for example, let us call it North/South direction) we see the robber going North with probability $1/4$, going South with the same probability and staying in place with probability $1/2$. In other words the robber performs a *lazy* random walk on the path. Hence, both coordinates behave similarly to the lazy random walk on integers (move with probability $1/2$; do nothing, otherwise). The same argument as in the previous proofs can be used to show that with probability, say, $1 - o(n^{-1})$, the robber stays within the distance $O(\sqrt{n\log n}) = o(n)$ from the initial vertex. Hence, if we look at the grid from a 'large distance' the drunk robber is not moving at all.

Therefore, since we would like to investigate an asymptotic behavior, the problem reduces to finding a set S consisting of two vertices such that the average distance to S is as small as possible. Cops should start on S to achieve the best outcome. It is clear that, due to the symmetry of $P_n\square P_n$, there are two symmetric optimal configurations for set S:

$$S = \{(n/2 + O(1), n/4 + O(1)), (n/2 + O(1), 3n/4 + O(1))\},$$
$$S = \{(n/4 + O(1), n/2 + O(1)), (3n/4 + O(1), n/2 + O(1))\}.$$

In any case, the average distance is

$$\sum_{u=0}^{n-1} \sum_{v=0}^{n-1} dist((u, v), S) = (1 + o(1))8n \int_{x=0}^{1/2} \int_{y=0}^{1/4} (x + y)\,dy\,dx = (1 + o(1))\frac{3}{8}n.$$

The result follows. □

4. The cost of drunkenness

In this section we show that the cost of drunkenness can be arbitrarily close to any real number $c \in [1, \infty)$. In order to do it, we introduce two families of graphs, barbells and lollipops.

4.1. Barbell

Let $n \in \mathbb{N}$ and $c \geq 0$. The *barbell* $B(n, c)$ is a graph that is obtained from two complete graphs $K_{\lfloor cn \rfloor}$ connected by a path P_n (that is, one end of the path belongs to the first clique whereas the other end belongs to the second one). The number of vertices of $B(n, c)$ is $(1 + 2c)n + O(1)$, and $c(B(n, c)) = 1$. In order to catch (either the adversarial or the drunk) robber, the cop should start at the center of the path and move toward the robber; $ct(B(n, c)) = n/2 + O(1)$. This family can be used to get any ratio from $(1, 2]$.

Theorem 4.1. *Let $c \geq 0$. Then,*

$$\text{dct}(B(n, c)) = (1 + o(1))\frac{n}{2} \cdot \frac{1 + 4c}{2 + 4c},$$

and the cost of drunkenness is

$$F(B(n, c)) = \frac{\text{ct}(B(n, c))}{\text{dct}(B(n, c))} = 1 + \frac{1}{1 + 4c} + o(1).$$

Proof. The drunk robber starts on a clique with probability $(2c)/(1 + 2c) + o(1)$. If this is the case, the capture occurs at time $n/2 + O(\sqrt{n \log n})$ with probability, say, $1 - o(n^{-1})$ by Lemma 2.3. If the robber chooses a vertex at the distance k from the robber to start with, he is captured after $k + O(\sqrt{n \log n})$ steps, again with probability $1 - o(n^{-1})$. Hence the expected capture time is

$$(1 + o(1)) \left(\frac{2c}{1 + 2c} \cdot \frac{n}{2} + \frac{1}{1 + 2c} \cdot \frac{n}{4} \right) = (1 + o(1))\frac{n}{2} \cdot \frac{1 + 4c}{2 + 4c}.$$

The theorem holds. □

4.2. Lollipop

Let $n \in \mathbb{N}$ and $c \geq 0$. The *lollipop* $L(n, c)$ is a graph that is obtained from a complete graph $K_{\lfloor cn \rfloor}$ connected to a path P_n (that is, one end of the path belongs to the clique). The number of vertices of $L(n, c)$ is $(1 + c)n + O(1)$, and the cop number $c(L(n, c))$ is 1. In order to catch the perfect robber, the cop should start at the center of the path and move toward the robber; $ct(L(n, c)) = n/2 + O(1)$. However, it is not clear what the optimal strategy for the drunk robber is. The larger the clique is, the closer to the clique the cop should start the game.

Theorem 4.2. *Let $c \geq 0$. Then,*

$$\text{dct}(L(n, c)) = \begin{cases} (1 + o(1))\frac{n}{4} \cdot \frac{(\sqrt{2} - 1 + c)(\sqrt{2} + 1 - c)}{1 + c}, & \text{for } c \in [0, 1] \\ (1 + o(1))\frac{n}{2(1 + c)}, & \text{for } c > 1. \end{cases}$$

and the cost of drunkenness is

$$F(L(n, c)) = \frac{\text{ct}(L(n, c))}{\text{dct}(L(n, c))} = \begin{cases} \frac{2(1 + c)}{(\sqrt{2} - 1 + c)(\sqrt{2} + 1 - c)} + o(1), & \text{for } c \in [0, 1] \\ (1 + c) + o(1), & \text{for } c > 1. \end{cases}$$

Before we move to the proof of this result, let us mention that the cost of drunkenness (as a function of the parameter c) has an interesting behavior. For $c = 0$ it is 2 (we play on the path), but then it is decreasing to hit its minimum of $1 + \sqrt{2}/2$ for $c = \sqrt{2} - 1$. After that it is increasing back to 2 for $c = 1$, and goes to infinity together with c. Therefore, this family can be used to get any ratio at least $1 + \sqrt{2}/2 \approx 1.71$.

Proof. Let the cop start on vertex v at the distance $(1 + o(1))bn$ from the clique ($b \in [0, 1]$ will be chosen to obtain the minimum expected capture time). The drunk robber starts on a vertex of a clique with probability $c/(1 + c) + o(1)$. If this is the case, the capture occurs at time $bn + O(\sqrt{n \log n})$ with probability, say, $1 - o(n^{-1})$ by Lemma 2.3. If the robber chooses a vertex at the distance k from the cop, then he is captured, again with probability $1 - o(n^{-1})$, after $k + O(\sqrt{n \log n})$ rounds. The robber starts between the cop and the clique with probability $b/(1 + c) + o(1)$ and on the other side with remaining probability. Hence the expected capture time is equal to

$$(1 + o(1)) \left(\frac{c}{1 + c} \cdot bn + \frac{b}{1 + c} \cdot \frac{bn}{2} + \frac{1 - b}{1 + c} \cdot \frac{(1 - b)n}{2} \right)$$

$$= (1 + o(1))\frac{n}{1 + c} \left(b^2 + (c - 1)b + 1/2 \right).$$

The above expression is a function of b (that is, a function of the starting vertex v for the cop) and is minimized at

$$b = \max \left\{ \frac{1 - c}{2}, 0 \right\}.$$

The theorem holds. □

It follows immediately from Theorems 3.4, 4.1 and 4.2 that the cost of drunkenness can be arbitrarily close to any constant $c \geq 1$.

Corollary 4.3. *For every real constant $c \geq 1$, there exists a sequence of graphs $(G_n)_{n \geq 1}$ such that*

$$\lim_{n \to \infty} F(G_n) = \lim_{n \to \infty} \frac{ct(G_n)}{dct(G_n)} = c.$$

5. Computational aspects

In this section we deal with computational aspects of the cop against drunk robber problem. Our analysis holds for any number of cops, that is, we no longer assume that $k = c\,(G)$.

5.1. Computing expected capture time for a given strategy

Suppose that we are given a graph and we fix a strategy before the game actually starts. We will now show how to explicitly compute the probability of capture at time $t \in \{0, 1, 2, \ldots\}$ as well as the expected capture time.

Fixing a strategy in advance is the best one can do for the invisible robber case (see Section 6), but for a visible one, cops should adjust their strategy based on the behavior of the opponent; this will be treated in the next Section 5.2. However, the approach presented here is less demanding computationally and can be used to provide an upper bound for the optimal expected capture time.

Let $G = (V, E)$ be a connected graph with $V = \{0, 1, \ldots, n - 1\}$. Letting

$$P_{i,j} = \mathbb{P}\,(Y_t = j \mid Y_{t-1} = i)$$

we have

$$P_{i,j} = \begin{cases} \frac{1}{|N(i)|} & \text{for } j \in N(i) \\ 0 & \text{otherwise.} \end{cases}$$

Note that P is the $n \times n$ transition probability matrix governing the robber's random walk in G *in the absence of cops*. To account for capture by the cops, define a new state space $\overline{V} = V \cup \{n\}$, that is, the old state space augmented by the *capture state n*. The corresponding $(n + 1) \times (n + 1)$ transition matrix is

$$\overline{P} = \begin{pmatrix} P & \mathbf{0} \\ \mathbf{0} & 1 \end{pmatrix}.$$

In the absence of cops, the robber performs a standard random walk on G and never enters the capture state; if however he starts in the capture state, he remains there forever: $\overline{P}_{n,n} = 1$. In other words, the Markov chain governed by $\overline{P}$ contains two noncommunicating equivalence classes: $\{0, 1, \ldots, n - 1\}$ and $\{n\}$.

Suppose now that a single cop is located in vertex x. We will denote the corresponding transition probability matrix by $\overline{P}\,(x)$. Obviously, $\overline{P}\,(x) \neq \overline{P}$. The difference is caused by the possibility of capture, which can occur in two ways.

(i) At the $(t - 1)$-th round the robber is located at x and, in the first phase of the t-th round, the cop moves into x. Then the robber is captured, so $\overline{P}_{x,n}\,(x) = 1$ and $\overline{P}_{x,y}\,(x) = 0$ for $y \in V$.

(ii) At the $(t - 1)$-th round the robber is located at $y \neq x$ and, in the second phase of the t-th round, he moves from y to x. Hence the robber is captured with probability $P_{y,x}$. So, for all $y \in V - \{x\}$, $\overline{P}_{y,n}\,(x) = P_{y,x}$, $\overline{P}_{y,x}\,(x) = 0$.

We can summarize the above by writing

$$\overline{P}\,(x) = \begin{pmatrix} P\,(x) & \mathbf{p}(x) \\ \mathbf{0} & 1 \end{pmatrix},$$

where $P\,(x)$ has 0's in the x-th row and column and the corresponding probabilities have been moved into the $\mathbf{p}(x)$ vector. For example, letting G be the path with 5 nodes, the matrices $\overline{P}$ and $\overline{P}\,(2)$ are:

$$\overline{P} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 1/2 & 0 & 0 \\ 0 & 0 & 1/2 & 0 & 1/2 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \qquad \overline{P}\,(2) = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Especially for the placement round of the game ($t = 0$) we need a different matrix, because the robber does not perform a random-walk, but simply chooses an initial position uniformly at random; if he chooses the one already occupied by the cop, then he is captured immediately. Hence, for this round the appropriate transition matrix is $\widehat{P}\,(x)$, which is the unit matrix with the one of the x-th row moved to the $(n + 1)$-th column.

Let $\pi_i(t) = \mathbb{P}(Y_t = i)$ for $i \in \overline{V}$ and $t \in \{0, 1, \ldots, s\}$ and $\pi(t) = (\pi_0(t), \pi_1(t), \ldots, \pi_n(t))$; also let $\widehat{\pi}(0) = \left(\frac{1}{n}, \frac{1}{n}, \ldots, \frac{1}{n}, 0\right)$. Then, given a strategy $\mathbf{X} = (x_0, x_1, \ldots, x_s)$, the above formulation yields

$$\pi(0) = \widehat{\pi}(0) \widehat{P}(x_0)$$

and, for $t \in \{1, 2, \ldots\}$,

$$\pi(t) = \pi(t-1) \overline{P}(x_{t-1}).$$

This implies that $\pi(t) = \widehat{\pi}(0) \widehat{P}(x_0) \overline{P}(x_1) \overline{P}(x_2) \ldots \overline{P}(x_t)$. To illustrate this, let us continue the example. Suppose a single cop enters the path and follows the strategy $\mathbf{X} = (0, 1, 2, 3, 4)$ (start on one end of the path and move to the other one). Then we have

$$\pi(0) = \widehat{\pi}(0) \widehat{P}(x_0) = \begin{pmatrix} 1/5 & 1/5 & 1/5 & 1/5 & 1/5 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \end{pmatrix}$$

$$\pi(1) = \pi(0) \overline{P}(x_1) = \begin{pmatrix} 0 & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 1/2 & 0 & 1/2 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} 0 & 0 & \frac{1}{10} & \frac{3}{10} & \frac{1}{10} & \frac{1}{2} \end{pmatrix}$$

$$\pi(2) = \pi(1) \overline{P}(x_2) = \begin{pmatrix} 0 & 0 & \frac{1}{10} & \frac{3}{10} & \frac{1}{10} & \frac{1}{2} \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} 0 & 0 & 0 & \frac{1}{10} & \frac{3}{20} & \frac{3}{4} \end{pmatrix}$$

$$\pi(3) = \pi(2) \overline{P}(x_3) = \begin{pmatrix} 0 & 0 & 0 & \frac{1}{10} & \frac{3}{20} & \frac{3}{4} \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

The elements $\pi_n(t)$ give the probabilities $\mathbb{P}(X_t = n)$ at time t, that is, the probabilities of capture in *at most* t steps. The probabilities of capture *exactly* at time t are then given by $\pi_n(t) - \pi_n(t-1)$. The expected capture time (conditional on strategy $\mathbf{X}$ being used) is

$$\mathbb{E}T = \sum_{t=1}^{\infty} t \cdot (\pi_n(t) - \pi_n(t-1)).$$

In the above example we have

$$\mathbb{E}T = 1 \cdot \left(\frac{1}{2} - \frac{1}{5}\right) + 2 \cdot \left(\frac{3}{4} - \frac{1}{2}\right) + 3 \cdot \left(1 - \frac{3}{4}\right) = \frac{31}{20}.$$

The approach can be generalized to more than one cop, by letting $\mathbf{x} = (x_1, x_2, \ldots, x_k)$ be a configuration of cops and defining $\overline{P}(\mathbf{x})$, $P(\mathbf{x})$ analogously to the one cop case. Given that the cops follow the strategy $\mathbf{X} = (X_1, X_2, \ldots, X_s)$, the transition probabilities of Y satisfy

$$\mathbb{P}(Y_t = j \mid Y_{t-1} = i) = P_{ij}(X_t)$$

for $t \leq s$. So the robber process is an inhomogeneous Markov chain, with the transitions controlled by the cops' actions. Markov chains of this type are called *Markov Decision Processes* (MDP) or *Controlled Markov Processes*, where the control function is X_t; it is a (stochastic) control in the sense that it allows us to change the transition probabilities of Y_t. We can use the MDP formulation to compute $\mathbb{E}T$ for any given strategy $\mathbf{X}$ in reasonable time. Computing the *optimal* strategy is not

computationally viable; for example, with $|V| = n$ and k cops there may exist up to $\Theta((n^k)^t)$ strategies of length t (and the same number of corresponding $\mathbb{E}T$'s) to evaluate. In the Section 5.2 we will present a computationally viable approach to compute the strategy that is arbitrarily close to the optimal one.

MDP's were introduced in the book [13]; book-length treatments are [3,21,25,27]; an online tutorial is [15]. They have been applied to a version of the cops–robber problem in [8].

5.2. Computing near-optimal strategies and minimum expected capture time

Let us now present an algorithm to compute $F(G) = \frac{\text{ct}(G)}{\text{dct}(G)}$ with arbitrarily good precision. Basically this reduces to computing $\text{ct}(G)$ and a good approximation of $\text{dct}(G)$, which can be done independently. To this end we present two algorithms, both of which have previously appeared in the literature. To improve the presentation we assign a name to each algorithm and make a few notational modifications; also we point out the similarity between the two algorithms (which apparently has not been noticed before).

(i) The *CAAR* (*C*op *A*gainst *A*dversarial *R*obber) algorithm computes $\text{ct}_{x,y}(G)$ for every initial cop/robber configuration (x, y). In addition, CAAR computes the optimal cop and robber play for every (x, y). Capture time $\text{ct}(G)$ is easily computed from $\text{ct}(G) = \min_x \max_y \text{ct}_{x,y}(G)$.

(ii) Similarly, the *CADR* (*C*op *A*gainst *D*runk *R*obber) algorithm computes (an arbitrarily good approximation of) $\text{dct}_{x,y}(G)$ and the (near-)optimal cop play for every (x, y); drunken capture time $\text{dct}(G)$ is computed from $\text{dct}(G) = \min_x \frac{\sum_y \text{dct}_{x,y}(G)}{n}$.

CAAR was introduced by Hahn and MacGillivray in [12]. We present the algorithm for the case of a single cop (the generalization for more than one cop is straightforward). Slightly changing notation, we will use $C_{x,y}$ to denote the game duration when the cop is located at x, the robber at y and it is the cop's turn to move (in other words, $C_{x,y}$ equals $\text{ct}_{x,y}(G)$). Similarly $R_{x,y}$ denotes the game duration when it is the robber's turn to move. For both $C_{x,y}$ and $R_{x,y}$ we assume optimal play by both cop and robber. Let us also define

$$\widehat{V}^2 = V \times V - \{(x, x) : x \in V\},$$

(that is, V^2 excluding the diagonal) and for all $x \in V$, let $N[x] = N(x) \cup \{x\}$ be the closed neighborhood of x. CAAR consists of the following recursion (for $i = 1, 2, \ldots$):

$$\forall (x, y) \in \widehat{V}^2 : R_{x,y}^{(i)} = \max_{y' \in N[y]} C_{x,y'}^{(i-1)}, \tag{1}$$

$$\forall (x, y) \in \widehat{V}^2 : C_{x,y}^{(i)} = 1 + \min_{x' \in N[x]} R_{x',y}^{(i)}. \tag{2}$$

C and R are initialized with $C_{x,y}^{(0)} = R_{x,y}^{(0)} = \infty$ for all $x \neq y$. We take $C_{x,x}^{(i)} = R_{x,x}^{(i)} = 0$ for $i = 0, 1, 2, \ldots$. Then (1)–(2) is essentially equivalent to the version presented by Hahn and MacGillivray in [12], with just one difference, which we will now discuss.

In (1)–(2) the matrix C is computed iteratively: the $(i - 1)$-th matrix $C^{(i-1)}$ is stored and used in the i-th iteration to compute $C^{(i)}$. In numerical analysis this is known as a *Jacobi* iteration. It is well known that an alternative approach to computations of this type is the *Gauss–Seidel* iteration. In this iteration a single copy of C is stored and its elements are updated "in place." In [12], Hahn and MacGillivray present the Jacobi version of CAAR and prove that the algorithm converges (in a finite number of steps) if and only if $c(G) = 1$. Hence CAAR computes the solution of the equations

$$\forall (x, y) \in \widehat{V}^2 : R_{x,y} = \max_{y' \in N[y]} C_{x,y'}, \tag{3}$$

$$\forall (x, y) \in \widehat{V}^2 : C_{x,y} = 1 + \min_{x' \in N[x]} R_{x',y}, \tag{4}$$

$$\forall x \in V : C_{x,x} = R_{x,x} = 0. \tag{5}$$

The interpretation of the equations is the following. Eq. (3) captures the property that from configuration (x, y) the robber moves so as to maximize the length of the game; similarly, (4) describes the cop's goal to minimize the game duration (since the cop moves in the first phase of each round, 1 time unit must be added to $\min R_{x',y}$); finally (5) says that the game ends when cop and robber occupy the same vertex.

Extending the CAAR idea to the *drunk* robber game, let us now use $C_{x,y}$ to denote $\text{dct}_{x,y}(G)$. In other words $C_{x,y}$ (respectively, $R_{x,y}$) is the *expected* game duration after the cop's (respectively, robber's) move. Recall (see Section 5.1) that $P_{y,y'}(x)$ is the probability of the robber transiting from y to y', given that the cop is at x; note that $P(x)$ is a *substochastic* matrix. The analog of (1)–(2) is

$$\forall (x, y) \in \widehat{V}^2 : R_{x,y}^{(i)} = \sum_{y' \in N(y)} P_{y,y'}(x) C_{x,y'}^{(i-1)}, \tag{6}$$

$$\forall (x, y) \in \widehat{V}^2 : C_{x,y}^{(i)} = 1 + \min_{x' \in N[x]} R_{x',y}^{(i)} \tag{7}$$

and the analog of (3)–(5) is

$$\forall (x, y) \in \widehat{V}^2 : R(x, y) = \sum_{y' \in N(y)} P_{y,y'}(x) C_{x,y'}, \tag{8}$$

$$\forall (x, y) \in \widehat{V}^2 : C_{x,y} = 1 + \min_{x' \in N[x]} R_{x',y}. \tag{9}$$

$$\forall x \in V : C_{x,x} = R_{x,x} = 0. \tag{10}$$

We want (6)–(7) to converge to the solution of (8)–(10). We will discuss convergence conditions (and initialization) presently.

Actually (6)–(7) can be simplified. Since the drunk robber does not choose his moves, we can eliminate $R_{x,y}^{(i)}$ from (6)–(7) and obtain the CADR algorithm recursion:

$$\forall (x, y) \in \widehat{V}^2 : C_{x,y}^{(i)} = 1 + \min_{x' \in N[x]} \left(\sum_{y' \in N(y)} P_{y,y'}(x') C_{x',y'}^{(i-1)} \right). \tag{11}$$

We have derived (11) from (6)–(7), which we see as an analog of (1)–(2). However, we will now show that (11) is a version of the *value iteration* algorithm, introduced and studied in the MDP literature [3,21,25,27]. Consider a general MDP process with state space S, action space A, transition matrix Q and cost matrix $G(a)$ (that is, $G_{s,s'}(a)$ is the cost of transition $s \to s'$ using action a). The state space satisfies $S = S_T \cup S_A$, where S_T are the transient states and S_A the absorbing ones; it is assumed that transitions after absorption have zero cost: $G_{s,s'}(a) = 0$ for $s, s' \in S_A$. Let C_s be the expected total cost of the process starting from state s and continuing until absorption. Then [25] C satisfies the equations

$$\forall s \in S_T : C_s = \min_{a \in A} \left(G_{s,s'}(a) + \sum_{s' \in S_T} Q_{s,s'}(a) C_{s'} \right) \tag{12}$$

and the solutions to (12) can be obtained by the following value iteration:

$$\forall s \in S_T : C_s^{(i)} = \min_{a \in A} \left(G_{s,s'}(a) + \sum_{s' \in S_T} Q_{s,s'}(a) C_{s'}^{(i-1)} \right). \tag{13}$$

To show that (13) can be reduced to (11) let us take $S_T = \widehat{V}^2$ and $A = V$; in other words, states $s = (x, y)$ are cop/robber configurations and actions $a = x'$ are new cop positions. Regarding move costs: (a) before capture every move has unit cost, (b) after capture only moves of the form $(x, x) \to (x, x)$ are possible and these have zero cost; in short

$$G_{(x,y),(x',y')}(x') = \begin{cases} 1 & \text{if and only if } x \neq y \\ 0 & \text{otherwise.} \end{cases}$$

Finally,

$$Q_{(x,y),(x',y')}(a) = \begin{cases} P_{y,y'}(x') & \text{if } a = x' \in N[x] \text{ and } y' \in N(y) \\ 0 & \text{otherwise.} \end{cases}$$

Using the above, it is easy to reduce (13) to (11).

The convergence of the CADR algorithm has been studied by several authors, in various degrees of generality [8,13,27]. A simple yet strong result, derived in [8], uses the concept of *proper strategy*: a strategy is called proper if it yields finite expected capture time. It is proved in [8] that: if a proper strategy exists for graph G, then the Gauss–Seidel version of CADR converges to the true C for arbitrary $C^{(0)}$ provided $C_{x,y}^{(0)} \geq 0$ for all $(x, y) \in \widehat{V}^2$. As we have seen in Theorem 2.1, the cop has a proper strategy for every G. It can be proved that the Jacobi version of CADR also converges under the same conditions.

Now, $F(G)$ can be computed, easily. For every pair (x, y), one can obtain a desired approximation of $\text{ct}_{x,y}(G)$ and $\text{dct}_{xy}(G)$ by performing CAAR and CADR, respectively. Then

$$F(G) = \frac{\text{ct}(G)}{\text{dct}(G)} = \frac{\min_{x \in V} \max_{y \in V} \text{ct}_{xy}(G)}{\min_{x \in V} \frac{1}{|V|} \sum_{y \in V} \text{dct}_{xy}(G)}.$$

Both CAAR and CADR can be generalized for the case of k cops, replacing x by a k-tuple $\mathbf{x} = (x_1, x_2, \ldots, x_k)$; however, the execution time of both algorithms increases exponentially with k, hence the algorithms are computationally viable only for small k's. Also CADR will work for any transition probability matrix P, not just for random walks. Hence, if desired, we can compute the cost of drunkenness for any number of cops (not just for $k = c(G)$) and for non-uniform random walks (i.e., discrete time birth-and-death processes) and other kinds of Markovian robbers.

Both CAAR and CADR can easily provide an optimal and near-optimal cop strategy in *feedback* form $U_{x,y}$, that is, the optimal cop move when the cop/robber configuration is (x, y). This is achieved by recording a minimizing x' in (4)/(11). The optimal robber strategy $W_{x,y}$ (for the adversarial robber) can be similarly obtained by CAAR. For every (x, y) configuration we can have more than one optimal move, but they all yield the same (optimal) game duration.

We have implemented the CAAR and CADR algorithms in the Matlab package `CopsRobber`, which can be downloaded from [16]. We have used this package to perform a number of numerical experiments, some of which are presented in the technical report [28]. This report also contains presentation of the algorithms in pseudo-code and a discussion of various computational issues.

6. The invisible robber

In this section we present an introductory discussion of the cops and robber game when the robber is *invisible*; in other words, the cops do not know the robber's location unless he is occupying the same vertex as one of the cops. All the other rules of the game remain the same. This version raises several interesting questions, a full study of which will be undertaken in a future paper.

Since the cops never see the robber until capture, they cannot use feedback strategies. In other words, the cop strategy is determined before the game starts. This does *not* mean that every cop move is predetermined, because in certain cases it makes sense for the cops to randomize their moves. Hence capture time will in general be a random variable, even in the case of adversarial robber (who may also benefit from a randomized strategy).

Let us first examine the case of adversarial invisible robber. It is clear that, given enough cops, expected capture time will be finite. This is obviously true for $|V|$ cops, but in fact $c(G)$ cops suffice, as seen by the following theorem.

Theorem 6.1. *Suppose that $c(G)$ cops perform a random walk on a connected graph G, starting from any initial position. The robber, playing perfectly, is trying to avoid being captured. Let random variable T be the capture time. Then,*

$$\mathbb{E}T < \infty.$$

Proof. Let $G = (V, E)$ be any connected graph, and let $\Delta = \Delta(G)$ be the maximum degree of G. Put $k = c(G)$. For any configuration of cops $x \in V^k$ and any vertex occupied by the robber $y \in V$, there exists a winning strategy $S_{x,y}$ that guarantees that the robber is caught after at most $t_{x,y}$ rounds. It is clear that cops will follow $S_{x,y}$ with probability at least $(1/\Delta)^{kt_{x,y}}$. Now, let us define

$$\varepsilon = \min_{x \in V^k, y \in V} (1/\Delta)^{kt_{x,y}} = (1/\Delta)^{kT_0} > 0, \quad \text{where } T_0 = \max_{x \in V^k, y \in V} t_{x,y}.$$

This implies that, regardless of the current position of players at time t, the probability that the robber will be caught after at most T_0 further rounds is at least ε. Moreover, corresponding events for times t, $t+T_0$, $t+2T_0$, ... are mutually independent. Thus, we get immediately that

$$\mathbb{E}T = \sum_{t \geq 0} \mathbb{P}(T > t) \leq \sum_{t \geq 0} \mathbb{P}\left(T > \left\lfloor \frac{t}{T_0} \right\rfloor T_0\right)$$

$$= \sum_{i \geq 0} T_0 \mathbb{P}(T > iT_0) \leq T_0 \sum_{i \geq 0} (1 - \varepsilon)^i = \frac{T_0}{\varepsilon} < \infty, \tag{14}$$

and we are done. □

Hence $c(G)$ is the *minimum* number of cops required to capture the adversarial invisible robber in finite expected time, since this task is at least as hard as capturing the adversarial *visible* robber. Of course, generally it will take longer, compared to the visible robber case, to capture the invisible robber. Let us define $ict_{x,y}(G, k)$ to be the expected capture time when the initial cops/robber configuration is (x, y) and both the k cops and the robber play optimally; we also define

$$ict(G, k) = \min_{x \in V^k} \max_{y \in V} ict_{x,y}(G, k)$$

and, finally, $ict(G) = ict(G, c(G))$.

We now turn to the *drunk* invisible robber. He chooses his starting vertex uniformly at random and performs a random walk, as before. For a given starting position $x \in V^k$ for k cops, there is a strategy that yields the smallest expected capture time $idct_x(G, k)$. Cops have to minimize this by selecting a good starting position:

$$idct(G, k) = \min_{x \in V^k} idct_x(G, k).$$

As usual, $idct(G) = idct(G, c(G))$ but it makes sense to consider any value of $k \geq 1$. The proof of the next theorem is exactly the same as Theorem 2.1 and so it is omitted.

Theorem 6.2. $idct(G, k) < \infty$ *for any connected graph G and $k \geq 1$.*

Finally, the cost of drunkenness for the invisible robber game is $F_i(G) = \frac{ict(G)}{idct(G)}$. It follows from the last theorem that this graph parameter is well defined (that is, finite).

Let us make a few remarks regarding the invisible robber with "*infinite*" speed (actually, what we mean by this is an arbitrarily high speed). Let us define the cop number for this case by $c^\infty(G)$; it is the minimum number of cops that have a strategy to obtain a finite expected capture time. It is clear that $c(G) \leq c^\infty(G) \leq s(G)$, where $s(G)$ is the *search number* of G, that is, the minimum number of cops required to *clean* the graph in the *Graph Search* (GS) game (mentioned in Section 1). We want to emphasize that the cops and robber game (with invisible, infinite speed robber) is *different* from the GS game and, in particular, there are graphs for which $c^\infty(G) < s(G)$. For example, for the C_3 cycle, $s(C_3) = 2$ but $c^\infty(G) = 1$, namely one cop using a *randomized* strategy, can capture the invisible, adversary, infinite speed robber in T with $\mathbb{E}T = 2$. Similarly, one cop on $K_{1,3}$, the star with 3 rays, can achieve $\mathbb{E}T = 11/3$. Many other examples can be found. The main reason for the discrepancy between $c^\infty(G)$ and $s(G)$ is that, in the GS game, the fugitive is assumed *omniscient* and (under one interpretation) this means he knows *in advance* all the cop moves (until the end of the game). In the cops and robber family of games, on the other hand, omniscience is not assumed, either explicitly or implicitly. We can summarize in one phrase: *cleaning is harder than capturing* even an infinite speed robber. We intend to further explore this issue, as well the computation of optimal strategies for cops chasing an invisible adversarial robber in a future publication.

We will finish this section with the computation of the cost of drunkenness for two examples (path and cycle) involving an invisible (unit speed) robber. In both cases the computation is possible because the optimal strategy (for both the cops and the adversarial robber) is "obvious". Our examples are similar to the ones we have considered for the visible robber and proofs are omitted, since they are almost identical to those of Section 3.

Consider the path P_n again, with a single cop and an invisible robber. It is clear that the best strategy for the cop (regardless of whether he is playing against a perfect robber or a drunk one) is to start from one end of the path (say, from vertex 0) and move along the path until the robber is captured. We have $\mathrm{ict}(P_n) = n - 1$. When cops are playing against a drunk robber, the expected capture time is roughly two times smaller.

Theorem 6.3.

$$\frac{n}{2}\left(1 - O\left(\frac{\log n}{n}\right)\right) \leq \mathrm{idct}(P_n) \leq \frac{n-1}{2}.$$

In particular, $\mathrm{idct}(P_n) = (1 + o(1))n/2$ *and the cost of drunkenness is*

$$F_i(G) = \frac{\mathrm{ict}(P_n)}{\mathrm{idct}(P_n)} = 2 + o(1).$$

Let us now play the game with two cops and an invisible robber on the cycle C_n for $n \geq 4$. It is not difficult to see that $s(C_n) = 2 = c(C_n)$. The best cop strategy is to start on vertices 1 and n; the cop occupying vertex 1 will move toward higher values, the other one will move in the opposite direction. The game ends after $\mathrm{ict}(C_n) = \lfloor (n-1)/2 \rfloor$ steps. When cops are playing against a drunk robber, the expected capture time is roughly two times smaller.

Theorem 6.4. *We have*

$$\frac{n}{4}\left(1 - O\left(\frac{\log n}{n}\right)\right) \leq \mathrm{idct}(C_n) \leq \frac{n-1}{4}.$$

In particular, $\mathrm{idct}(C_n) = (1 + o(1))n/4$ *and the cost of drunkenness is*

$$F_i(G) = \frac{\mathrm{ict}(C_n)}{\mathrm{idct}(C_n)} = 2 + o(1).$$

7. Conclusion

Most of the results in the paper pertain to the case of a visible (adversarial/drunk) robber, pursued by $k = c(G)$ cops. The cases of arbitrary k and invisible robber have been briefly touched. We conclude the current paper by listing additional questions regarding the cost of drunkenness. We begin by listing several questions related to the visible robber.

 (i) Our analysis can be expanded to strategies which use an arbitrary number of cops. As shown in Theorem 2.1, even a single cop can catch a drunk robber in finite expected time. Hence, for a given G we can study $\mathrm{dct}(G, k)$ as a function of k. Obviously this is a decreasing function; what more can be said about it? As a first step in this direction, the numerical approach of Section 5 can be used to explore the properties of $\mathrm{dct}(G, k)$ for a given graph G.
 (ii) Let us define $\mathrm{dct}(G, \mathbf{X})$ to be the expected capture time in graph G using strategy $\mathbf{X}$. It is no longer assumed that $\mathbf{X}$ is an optimal strategy. Under what conditions on $\mathbf{X}$ and/or G will $\mathrm{dct}(G, \mathbf{X})$ be finite? Can we use the approach of Section 5 to obtain non-trivial bounds on $\mathrm{dct}(G, \mathbf{X})$?
(iii) A related question is whether (for a specific G and either optimal or general strategies) expected capture time can be connected to some graph parameter such as treewidth, pathwidth etc.

(iv) How robust are our results to slight (natural) modifications of the cops/robber game rules? For example, would the cost of drunkenness change if we allowed the robber to loop into its current location (that is, to perform lazy random walk)? What about a "general" random walk (that is, with nonuniform transition probabilities). What about *directed* graphs? Finally, does the situation change significantly if the cops and the robber move simultaneously rather than the cops moving first? The algorithm of Section 5 can be easily modified to handle these cases and numerical experiments may be useful for an initial exploration.

One can try to obtain similar results for the *invisible* robber. In Section 6 we showed how our approach can be extended (at least for certain families of graphs) to this case. In the examples we examined (paths, cycles) the optimal cop strategy is obvious. For general graphs, finding the search strategy which is optimal for the invisible (adversarial/drunk) robber will be more complicated. Is there a (computationally viable, perhaps approximate) algorithm to achieve this?

Finally, let us note that all of the above analyses adopt the cops' point of view. It will be interesting to study the cost of drunkenness for the cops. In other words, assuming an adversarial evader and k drunk cops, can we place bounds on the increase of expected capture time as compared to the case of adversarial cops? Theorem 6.1 may be used as a starting point to achieve this goal.

References

 [1] M. Aigner, M. Fromme, A game of cops and robbers, Discrete Applied Mathematics 8 (1984) 1–12.
 [2] B. Alspach, Sweeping and searching in graphs: a brief survey, Matematiche 59 (2006) 5–37.
 [3] J. Bertsekas, J. Tsitsiklis, Parallel and Distributed Computation, Addison-Wesley, 1989.
 [4] B. Bollobás, G. Kun, I. Leader, Cops and robbers in a random graph, preprint.
 [5] A. Bonato, R. Nowakowski, The Game of Cops and Robbers on Graphs, AMS, 2011.
 [6] A. Bonato, P. Prałat, C. Wang, Network security in models of complex networks, Internet Mathematics 4 (2009) 419–436.
 [7] D. Coppersmith, P. Tetali, P. Winkler, Collisions among random walks on a graph, SIAM Journal on Discrete Mathematics 6 (1993) 363–374.
 [8] J.H. Eaton, L.A. Zadeh, Optimal pursuit strategies in discrete-state probabilistic systems, Transactions of the ASME. Series D, Journal of Basic Engineering 84 (1962) 23–29.
 [9] F.V. Fomin, D. Thilikos, An annotated bibliography on guaranteed graph searching, Theoretical Computer Science 399 (2008) 236–245.
[10] P. Frankl, Cops and robbers in graphs with large girth and Cayley graphs, Discrete Applied Mathematics 17 (1987) 301–305.
[11] G. Hahn, Cops, robbers and graphs, Tatra Mountain Mathematical Publications 36 (2007) 163–176.
[12] G. Hahn, G. MacGillivray, A note on k-cop, l-robber games on graphs, Discrete Mathematics 306 (2006) 2492–2497.
[13] R.A. Howard, Dynamic Programming and Markov Process, MIT Press, 1960.
[14] S. Janson, T. Łuczak, A. Ruciński, Random Graphs, Wiley, New York, 2000.
[15] L. Kallenberg, Markov Decision Processes, http://www.math.leidenuniv.nl/~kallenberg/Survey%20MDP.pdf.
[16] Ath. Kehagias, P. Prałat, Cops and visible robbers, Technical Report available at http://users.auth.gr/~kehagiat/GraphSearch/TRCODvis.pdf.
[17] T. Łuczak, P. Prałat, Chasing robbers on random graphs: zigzag theorem, Random Structures and Algorithms 37 (2010) 516–524.
[18] A. Mehrabian, The capture time of grids, Discrete Mathematics 311 (2011) 102–105.
[19] S. Neufeld, R. Nowakowski, A game of cops and robbers played on products of graphs, Discrete Mathematics 186 (1998) 253–268.
[20] R. Nowakowski, P. Winkler, Vertex to vertex pursuit in a graph, Discrete Mathematics 43 (1983) 230–239.
[21] R. Pallu de la Barriere, Optimal Control Theory, Dover, 1980.
[22] P. Prałat, When does a random graph have constant cop number?, Australasian Journal of Combinatorics 46 (2010) 285–296.
[23] P. Prałat, N. Wormald, Meyniel's conjecture holds for random d-regular graphs, preprint.
[24] P. Prałat, N. Wormald, Meyniel's conjecture holds for random graphs, preprint.
[25] M.L. Puterman, Markov Decision Processes, Wiley, 1994.
[26] A. Quilliot, Jeux et pointes fixes sur les graphes, Ph.D. Dissertation, Université de Paris VI, 1978.
[27] D.J. White, Markov Decision Processes, Wiley, 1993.
[28] http://users.auth.gr/~kehagiat/GraphSearch/.