Discrete Mathematics 339 (2016) 1677–1681

Contents lists available at ScienceDirect

Discrete Mathematics

journal homepage: www.elsevier.com/locate/disc

Cop vs. Gambler

Natasha Komarov [a,*], Peter Winkler [b]

[a] Department of Mathematical Sciences, St. Lawrence University, Canton, NY 13617, USA

[b] Department of Mathematics, Dartmouth College, Hanover NH 03755-3551, USA

ARTICLE INFO

Article history:
Received 30 June 2015
Received in revised form 22 January 2016
Accepted 23 January 2016
Available online 17 February 2016

Keywords:
Pursuit/evasion games
Cops and robbers

ABSTRACT

We consider a variation of cop vs. robber on graph in which the robber is not restricted by the graph edges; instead, he picks a time-independent probability distribution on $V(G)$ and moves according to this fixed distribution. The cop moves from vertex to adjacent vertex with the goal of minimizing expected capture time. Players move simultaneously. We show that when the gambler's distribution is known, the expected capture time (with best play) on any connected n-vertex graph is exactly n. We also give bounds on the (generally greater) expected capture time when the gambler's distribution is unknown to the cop.

1. Introduction

The game of cops and robbers on graphs was introduced independently by Nowakowski and Winkler [13] and Quilliot [14], and has generated a great deal of study in the three decades since; see, e.g., [3–5,10,11]. In the original formulation a cop and robber move alternately from vertex to adjacent vertex (or stay where they are) on a connected, undirected graph G. The players have full information about each other's current position at each step. The cop's goal is to minimize capture time, the robber's to maximize it. There are graphs on which a robber playing optimally can elude the cop forever; for instance, chasing the robber on the 4-cycle is clearly a hopeless endeavor for the cop. Graphs on which a cop can win are called "cop-win". More precisely, a graph is cop-win if there is a vertex u such that for every vertex v, the cop beginning at u can capture the robber beginning at v. Cop-win graphs – also known as "dismantlable" graphs [13] – have appeared in statistical physics [6,7] as well as combinatorics and game theory.

The capture time in the original version of the game played on a cop-win graph has been analyzed and found to be at most $n-4$ for all graphs with $n \geq 7$ vertices [4,9]. This game contains equitable restrictions on the movements of the two players: the cop and robber are both constrained by the graph and can both see each other. What happens to the capture time if the rules are asymmetrical, and/or the game is played "at night"? In the "hunter and rabbit" game [1,2], the players move without seeing each other, and the robber-turned-rabbit is not constrained by the graph edges; that is, he is free to move to any vertex of the graph at each step. It turns out that the rabbit has a strategy that will get him expected capture time $\Omega(n \log n)$ on the n-cycle (or any graph of linear diameter).

Here we consider a pursuit game with the following rules. The game is played on a graph G (which will be assumed throughout this work to be connected and undirected with no loops or multiple edges) with $V(G) = \{v_1, v_2, \ldots, v_n\}$. The cop is constrained to the graph as above, moving from vertex to adjacent vertex or staying put at each step. The robber, whom we will now call a **gambler**, chooses a probability distribution $p_1, p_2, \ldots, p_n$ on $V(G)$ so that at each time $t \geq 0$, he is at vertex v_i with probability p_i. We call this probability distribution his **gamble**. The players move simultaneously in the dark and the game continues until the gambler is captured—that is, until the players occupy the same vertex at the same

* Corresponding author.
E-mail addresses: nkomarov@stlawu.edu (N. Komarov), peter.winkler@dartmouth.edu (P. Winkler).

http://dx.doi.org/10.1016/j.disc.2016.01.015
0012-365X/

time. The **capture time** is the number of moves up to and including the capture, thus a positive integer. The cop's objective is to minimize expected capture time while the gambler does his best to maximize it; thus we may think of the expected capture time, with best play by both players, as the **value** of the game (to the gambler).

We consider two variations: one in which the cop knows the gamble, and one in which she does not. When the gamble is known by the cop, we have the following rather surprising result: the value of the cop vs. gambler game is exactly n regardless of the graph structure or the cop's initial position!

Pursuit games have obvious application in warfare (e.g., destroyer vs. submarine) and crime-fighting, but our somewhat less adversarial cop vs. gambler game is perhaps more likely to appear in software design. Imagine, for example, that an anti-incursion program has to navigate a linked list of ports, trying to minimize the time to intercept an enemy packet as it arrives. If the enemies' port-choice distribution is known we get a version of the known gambler, otherwise the unknown gambler.

2. Cop & gambler on a tree

We suppose first that the graph G on which the game is played is a tree with vertices $v_1, \dots, v_n$, with the cop beginning at vertex v_1 (which we think of as the root).

Lemma 2.1. *The cop can capture the gambler in expected time at most n on any tree of order n.*

Proof. For any i and j, we let P_{ij} be the (unique) path from v_i to v_j. We denote by B_i the branch of G beginning at v_i, that is, $B_i := \{v_j : v_i \in V(P_{1j})\}$. Let $m_i := |B_i|$ be the number of vertices in that branch and $c_i := \sum_{v_j \in B_i} p_j$ the sum of the probabilities assigned to that branch by the gamble.

We will now (re)-number the vertices of G so that the cop's strategy will be to follow the path $v_1, v_2, \dots, v_k$ from the root toward a leaf, possibly stopping for good at some vertex on the way. The path is defined inductively as follows: given $v_1, \dots, v_i$, let v_{i+1} be a neighbor of v_i, other than v_{i-1}, that maximizes c_{i+1}/m_{i+1}. (Informally, the cop enters a branch with vertices of highest average probability.) If there is no such u, i.e., if v_i is a leaf with $i > 1$, then $k = i$ and the path-labeling is finished; the remaining vertices of G are numbered arbitrarily.

Let T_i be the expected capture time (always assuming best play) from the moment the cop moves to v_i; we will prove, by backward induction on i, that if v_i is reached, then $T_i \le m_i/c_i$. Note that $m_1 = n$ and $c_1 = 1$, thus the claim is equivalent to the statement of the lemma for $i = 1$.

If the cop reaches the leaf v_k, she remains there and since capture is now a matter of waiting for success in a sequence of i.i.d. Bernoulli trials with success probability p_k, we have $T_k = 1/p_k = m_k/c_k$, establishing the base of the induction.

Suppose the cop is at v_i, $i < k$, and that the claim holds for all $i \ge k$. If $p_i \ge c_i/m_i$, the cop stays at v_i and captures in expected time $1/p_i \le m_i/c_i$ as claimed. Otherwise she moves on to v_{i+1} giving

$$T_i \le 1 + (1 - p_i)T_{i+1} \le 1 + (1 - p_i)\frac{m_{i+1}}{c_{i+1}}$$

by the induction assumption.

Since the average probability of vertices in $B_i \backslash \{v_i\}$ is $\frac{c_i - p_i}{m_i - 1}$, and v_{i+1} was chosen to maximize average probability in B_{i+1}, we know that $\frac{c_{i+1}}{m_{i+1}} \ge \frac{c_i - p_i}{m_i - 1}$. Hence,

$$T_i \le 1 + (1 - p_i)\frac{m_i - 1}{c_i - p_i} = 1 + (m_i - 1)\frac{1 - p_i}{c_i - p_i}.$$

Noting that $\frac{1-p_i}{c_i - p_i}$ decreases as p_i decreases, and recalling that $p_i < c_i/m_i$, we deduce that

$$T_i \le 1 + (m_i - 1)\frac{1 - c_i/m_i}{c_i - c_i/m_i}.$$

And now, noting that $m_i > 1$ and $c_i > 0$, we get:

$$\begin{aligned} T_i &\le 1 + (m_i - 1)\frac{1 - c_i/m_i}{c_i - c_i/m_i} \\ &= 1 + \frac{(m_i(m_i - 1) - c_i(m_i - 1))/m_i}{(m_i c_i - c_i)/m_i} \\ &= 1 + \frac{m_i(m_i - 1) - c_i(m_i - 1)}{m_i c_i - c_i} \\ &= \frac{c_i(m_i - 1) + m_i(m_i - 1) - c_i(m_i - 1)}{c_i(m_i - 1)} \\ &= \frac{m_i}{c_i} \end{aligned}$$

and the proof is complete. □