CAPTURE TIME IN VARIANTS OF COPS & ROBBERS GAMES

A Thesis

Submitted to the Faculty

in partial fulfillment of the requirements for the

degree of

Doctor of Philosophy

in

Mathematics

by

Natasha Komarov

DARTMOUTH COLLEGE

Hanover, New Hampshire

July 30, 2013

Examining Committee:

Peter Winkler, Chair

Richard Nowakowski

Sergi Elizalde

Peter Doyle

F. Jon Kull, Ph.D.
Dean of Graduate Studies

Abstract

We examine variations of cops and robbers games on graphs. Our goals are to introduce some randomness into their study, and to estimate (expected) capture time. We show that a cop chasing a random walker can capture him in expected time $n + o(n)$. We also discuss games in which the players move in the dark (showing that a cop can capture an immobile hider in time n on any graph and any robber in time n on K_n) and in which the players suffer various restrictions on their movements. Finally, we consider open problems, including the idea of a patrolling scheme—that is, a plan for the "beat" a cop ought to walk on a graph in order to maximize the danger for the robber of committing a crime at any given location.

Acknowledgements

I want to thank all of the people that made it possible for me to get through graduate school. First of all, thank you to my parents: without your lifelong support I probably wouldn't have been in graduate school to begin with (an obvious understatement—I would not have been here in many senses of the phrase without you). Thank you Asa: you have always kept things running so smoothly while I worried about this thesis and the work it contains that it's taken me a few years to understand how amazing of a partner you actually are; you've also changed for the better my entire mental layout about life, math, and everything. Thank you Misha and Oliver, for being there.

Thank you to my advisor, Peter Winkler: you've been a great advisor, and I don't think there's anything more for which I could have asked. Thank you also to the rest of my thesis defense committee: Richard Nowakowski, Peter Doyle, and Sergi Elizalde. And finally, thank you to all of the people I've spoken with over the years who have given me ideas, directly or indirectly, that have fed into my work. In particular, the work in Chapter 2 benefited from conversations at Microsoft Research in Redmond, Washington with Omer Angel, Ander Holroyd, Russ Lyons, Yuval Peres, and David Wilson.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The game of cops and robbers on graphs was introduced independently by Nowakowski and Winkler [41] and Quilliot [44], and has generated a great deal of study; see, e.g., [8, 11, 25, 27]. In the original formulation a cop and robber move alternately and deliberately, with full information, from vertex to adjacent vertex on a (finite, connected, undirected, and simple) graph G, with the cop trying to capture the robber and the robber trying to elude the cop. The game continues until the cop has captured the robber, if such a time occurs. A graph is said to be "cop-win" if there is a vertex u such that for every v, the cop beginning at u can capture the robber beginning at v in finite time. Otherwise it is "robber-win." In addition to their obvious role in pursuit games, cop-win graphs—also known as "dismantlable" graphs—have appeared in diverse places including statistical physics [13, 14].

Much work has been done in the study of the "cop number" of a graph [5, 8, 17]—that is, the number of cops necessary in order to make the graph cop-win. For more on the cop number, the reader is invited to read about Meyniel's conjecture [21] and, for instance, the study of cop number on special cases of graphs [33], in games in which the robber may have a different speed than the cop [20, 22, 40], in which capture may occur from a distance [10], and many other variants. These results apply in the deterministic version of the game originally presented—generally, games are played with full visibility. We are largely interested in variations involving lacking information in some way.

For instance, what happens in the version of the game in which the robber is moving according to a random walk? Then one cop will certainly suffice (see the introduction of Chapter 2), but now the question we would like to answer is how long this game takes (in expected time)—and indeed find that the answer is still around n (as it is in the original deterministic formulation). The question of capture time in the original cops and robbers game has been thoroughly considered on undirected graphs [8, 24]. For more on capture time in the original variation, including on more general graphs, see Chapter 6.

What if both players remain in control of their fates and are free to choose their respective strategies, but the games are played in the dark? The same question interests us again. In the work that follows, we will discuss several different and independent variations of this game, with varying restrictions being placed on either player. However, we are (nearly) always going to be bound by this unifying theme posed by the question—"How long does it take the cop to win?" Sometimes, this will mean that we are considering a class of graphs on which the cop can win (an analog of the dismantlable graphs in the original formulation to an alternate version of the game, as in Chapter 4) and seeing how long this can take in the worst case. Much of the time, we are going to be more interested in bounding the expected capture time in the case of a game whose set-up allows a single cop to win with probability 1.

1.1 History of Pursuit and Evasion Games

"Cops and Robbers" is an example of a class of games called "pursuit and evasion." The study of pursuit and evasion type games has its roots in the mathematical study of military strategy dating back to destroyer vs. submarine problems during World War I and later, missile guidance systems in the 1950's ([30]). These were generally set in a continuous-time setting. In 1976, Torrence Parsons first described a variant of pursuit-evasion played on a graph [42]. Though there are many variations, they are generally classified under the headings of continuous pursuit-evasion as in the geometric formulation (e.g. hunter and rabbit [1], princess and monster [30], the homicidal chauffeur problem [30], the Apollonius pursuit problem [31], and many others) and discrete pursuit-evasion: the graph theoretic formulation. In the present work, we will primarily be interested in the latter.

There are a number of applications of pursuit and evasion games. In addition to the military work from which the concept originated, there are a number of applications in robotics (such as collison-avoidance [30], air-traffic control [7], and surveillance [30]) and even search-and-rescue: in graph searching, we often have a searcher (who acts much as the pursuer in a pursuit and evasion game) and a usually invisible hider (one who may be mobile or immobile). For more on graph searching, see [3, 4, 19] among others. An intimately related field is that of graph cleaning (see [35] and [39] for the original formulation of this game). Graph cleaning generally consists of pipes (edges) that must be cleaned of some contaminant (perhaps a contaminant that can regenerate), and of brushes that sit at vertices and can be dispersed along the pipes to clean them. The usual interest in graph cleaning is to determine the minimal number of brushes needed to clean a graph. For some other examples of graph cleaning, see, e.g. [2, 37, 38].

1.2 Assumptions and Notation

1.2.1 General assumptions

We will assume unless otherwise noted that all games are played on a graph G which is finite, undirected, connected, and simple (that is, it contains no loops or multiple edges). When not stated otherwise, any discussion of random walks may be assumed to be about non-lazy random walks—that is, the walker is forced to move at each step (and given the lack of loops in the graph on which he is walking, this means he changes his location at each step).

We will have differing assumptions in various chapters about the progress of the game in question. We will therefore point out in each chapter whether the players' initial positions are known, whether their movements are visible to the other player, and whether they move alternately or simultaneously. For a summary of the major assumptions considered, please see the organizational chart in Section 7.1.

1.2.2 Notation

We will suppose that the graph G on which the games take place have vertex set $V(G)$ and edge set $E(G)$, where $|V(G)| = n$ and $|E(G)| = m$. The maximum degree of G is denoted $\Delta(G)$ (or Δ where there is no possibility of confusion) and for any vertex $v \in V(G)$, $\deg(v) = |N_G(v)|$ denotes the degree of v (i.e. the number of neighbors v has). We use the notation $\{1..n\}$ to denote the integers between 1 and n (inclusive). For any $t \geq 0$, by c_t and r_t we will denote the positions of the cop/hunter and the

1.2 Assumptions and Notation

adversary (robber/mole/gambler/drunk), respectively.

Chapter 2

Cops & Drunks

We now consider a variation suggested by Ross Churchley of the University of Victoria [34], in which the robber is no longer in control of his fate; instead, at each step he moves to a neighboring vertex chosen uniformly at random. We may therefore imagine that the robber is in fact a drunk—one who is too far gone to have an objective. As in the original game, the players move alternately and with full information (though this information is of course of no use to our inebriated adversary). In this variation, a "move" (as in chess) will consist of a step by the cop followed by a (uniformly random) step by the drunk. Capture or "arrest" takes place when the cop lands on the drunk's vertex or vice-versa, and the capture time T is the number of the move at which this takes place.

On any graph, the drunk will be caught with probability one, even by a cop who oscillates on an edge, or moves about randomly; indeed, by any cop who isn't actively trying to lose. The only issue is: how long does it take? The lazy cop will win in expected time at most $4n^3/27$ (plus lower-order terms), since that is the maximum possible expected hitting time for a random walk on an n-vertex graph [12]; the same bound applies to the random cop [16]. It is easy to see that the greedy cop who merely moves toward the drunk at every step can achieve $O(n^2)$; in fact, we will show that the greedy cop cannot in general do better. Our smart cop, however, gets her man in expected time $n+o(n)$. Note that when the adversaries play on a lollipop graph consisting of a clique of size $cn^{1/3}$ (for some constant $c \in \mathbb{R}$) with a path of length $n-cn^{1/3}$ attached at one end, with the drunk starting in the clique and the cop starting at the opposite endpoint of the path, the expected capture time will be $n - \Theta(n^{1/3}) = n - o(n)$, and we conjecture that this is worst possible.

2.1 Preliminaries

Let us consider some examples. (1) Suppose G is the path P_n on n vertices. Then the cop will (using any of the algorithms we consider later) move along the path until she reaches the drunk; this will take expected time about $n-\sqrt{n}$ since a random walk on a path will on average progress about distance $\sqrt{t}$ in time t.

(2) Let G be the complete balanced bipartite graph $K_{\lfloor n/2 \rfloor, \lceil n/2 \rceil}$, with the cop and the drunk beginning on the same side. Then the poor cop will find herself always moving to the opposite side from her quarry until, finally, he runs into her; since the latter event occurs with probability about $2/n$, arrest takes on average $n/2$ steps.

The reader may feel with some justification that we are being unrealistic in not allowing the cop to stay put; in example (2), sitting for one move would enable her to catch the drunk on the next move. Ultimately, we force the cop to move at each step in order to hold her to the same constraints as her quarry's, and because it gives us the strongest results. Our bounds still apply when the cop, the drunk, or both are allowed to stay put on any move.

2.1 Preliminaries

Even when the cop is permitted to idle, she cannot expect to catch the drunk in time bounded by a function of the diameter of G. For example (3), let G be the incidence graph of a projective plane of order n. A projective plane P of order n is a collection of objects called "points" and sets of points called "lines" satisfying the following conditions:

(a) Two points determine a unique line.

(b) Two lines intersect in a unique point.

(c) Every line consists of exactly $n+1$ distinct points.

(d) Every point lies on exactly $n+1$ distinct lines.

Furthermore [28],

(e) P contains exactly $n^2 + n + 1$ distinct points.

(f) P contains exactly $n^2 + n + 1$ distinct lines.

Projective planes of order n are known to exist for $n = p^a$ for any prime number p and positive integer a [15]. The incidence graph G of P is therefore a graph with $2(n^2 + n + 1)$ vertices, with adjacency relation $u \sim v$ if u is a point in P and v is a line that goes through u, or vice versa. Such graphs have bounded diameter but unbounded expected capture time:

Claim 1. $\mathrm{diam}(G) = 3$.

Proof. Let a, b be two points in P. By condition (1) above, a and b both lie on a common line, so $d(a, b) = 2$. If a, b are instead two lines in P, then condition (2) says that a and b intersect at a common point. Finally, if a is a point and b is a line in P, then either a lies on b and so $d(a,b)=1$ or there is another point, c, which does lie on b. But by the previous argument, $d(a,c)=2$ and so $d(a,b)=3$. □

Claim 2. The girth of G is 6.

Proof. Note that G has no odd cycles by the independence of the set of points (and respectively, set of lines). Now assume for sake of contradiction that G contains a cycle of length 4. Then there are two points p_1, p_2 and two lines ℓ_1, ℓ_2 such that $p_1, \ell_1, p_2, \ell_2, p_1$ forms a cycle. But this contradicts condition (2) since ℓ_1 and ℓ_2 must intersect in p_1 as well as p_2. □

Claim 3. G is regular of degree $r = n+1$ (approximately $\sqrt{|V(G)|/2}$).

Proof. By conditions (3) and (4). □

Claim 4. The expected capture time on G is at least r.

Proof. When the cop gets to distance 2 away from the drunk, the drunk has only one bad move out of r; the rest keep him at distance at least 2. (Similarly, if the cop gets to distance 1, bypassing ever being at distance 2, the drunk still has only one bad move out of r, the rest of which keep him at distance 1.) Hence the cop's expected capture time cannot be any lower than r (the expected number of independent Bernoulli trials, each with success probability $1/r$, until success is achieved). □

On the other hand, it is not hard to verify that on any regular graph, the greedy cop—who minimizes her distance to the drunk at each move—wins in expected time at most linear in n. If G is regular of degree r, its diameter cannot exceed $\frac{3n-r-3}{r+1}$ [45]. Since the drunk will step toward the cop with probability at least $1/r$ at each move, resulting (after her response) in a decrease of 2 in their distance, the expected capture time is bounded by $r \cdot \text{diam}(G)/2 < 3n/2$.

The linear bound also holds on trees. To see this, we proceed by induction on the size of the tree, n. When $n = 2$, the capture time is clearly less than n (since the drunk will run into the cop on his first move). Now suppose that on any tree with $t < n$ vertices, the expected capture time is at most t, and let T be a tree on n vertices, rooted at c_0 (the cop's initial position). For all descendants v of c_0, let T_v be the subtree of T consisting of v and all of its descendants. So the game begins on $T = T_{c_0}$, and after the first move, since the drunk cannot get "behind" the cop without being caught, the game is being played on T_{c_1} where c_1 is the cop's position after one step. (Note that by the greedy strategy, c_1 is the unique neighbor of c_0 which is on the path from c_0 to r_1, the drunk's position after he takes his first step.) $|V(T_{c_1})| \leq |V(T_{c_0})| - 1 = n - 1$ so by the induction hypothesis, the game takes expected time less than $n-1$ on T_{c_1} and therefore less than n on T.

For general G, one can guarantee only that at a given point in time the drunk will step toward the cop with probability at least $1/\Delta$, giving a bound of order n^2 for the greedy cop. That may appear to be a gross overestimate, especially in light of the special cases discussed above, but a graph with many high-degree vertices can still have large diameter. For example (4), consider the following graph.

The "ladder" in this graph consists of two copies of the path $P_{n/4}$ with each pair of

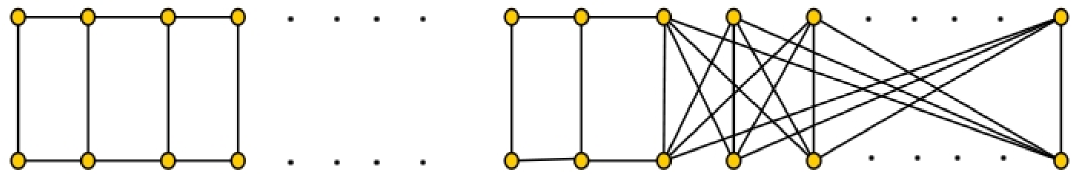

Figure 2.1: The Ladder to the Basement

corresponding vertices connected by an edge. The "basement" consists of a complete bipartite graph, $K_{\lfloor n/4 \rfloor, \lceil n/4 \rceil}$. We begin with the drunk inside the basement, and the cop on the far end of the ladder. While the drunk is meandering inside the basement, the cop—staying true to her goal of minimizing the distance between her and the drunk at each step—is alternating between the two paths. Note that we assume she makes the foolish choice when she is presented with several options by her algorithm. It takes the drunk $n/4$ moves on average to leave the basement, and each time this occurs, the cop will decrease the distance by 2 by traveling along her current path. Therefore the capture will require an average of about $(n/4)^2/2$ steps.

2.2 The Smarter Cop

2.2.1 Intuition

As noted in example (4) with the "ladder to the basement" graph, a foolish greedy cop can be foiled by her desire to "retarget" too often. That is, since she updates the target vertex (to which she is trying to minimize her distance) at each step, she is made indecisive by an indecisive drunk. One natural solution to this problem would be to walk directly toward the robber's initial position in the basement for several steps before retargeting. Continuing in this way, the cop makes steady progress, ultimately catching the drunk in time less than n.

In general, if a cop and drunk begin at distance d on a graph, and the cop proceeds by retargeting every four steps, then by Lemma 2.2 below, it would take

$$4(4n^{2/3})(d-3) \tag{2.1}$$

moves to get down to distance less than four. Since d can be as large as $n-1$, this would not suffice to yield our promised bound of $n+o(n)$, so the cop must first do something else to get her distance to the drunk down without spending too much time doing so—hence the following four-stage strategy.

For $i \in [4]$, let T_i be the time spent in Stage i and D_i be the distance between the two players at the end of Stage i. In the first stage, the cop heads directly for the drunk's initial position, x, so that $T_1 \leq \operatorname{diam}(G)$. In the meantime, the drunk has gone somewhere else, and so suppose that by the time that the cop reaches x, the drunk is at y. Now we are in Stage 2, and the cop heads for y. We show $\mathbb{E}[T_2] = o(n)$. During Stage 3, the cop updates her "target" every four steps, and we show that the expected time for this stage, $\mathbb{E}[T_3]$, is again bounded by $o(n)$. This stage ends when we are at distance at most three from the drunk. In Stage 4, the cop waits for the drunk to make an error, which happens in expected time at most Δ and results in the capture of the drunk. All together, this cop captures the drunk in expected time $n + o(n)$. We will refer to the progress made by the cop in the first two stages as "gross progress," and in the last two stages as "fine progress." In order to prove the bounds claimed above, it will be beneficial to have a few lemmas.

2.2.2 Gross Progress

Suppose that the drunk starts on vertex u and the cop starts at v. As noted in the set-up of the previous section, in the first stage of the cop's strategy, she is concerned only with getting to u (even if this may not decrease her distance from the drunk at the end of the stage). Clearly the time this takes is equal to $T_1 = d(v, u) \leq \text{diam}(G)$. We would like to get a bound on $\mathbb{E}[D_1]$, the expected distance between the cop and the drunk at the end of this stage. For that, the following lemma will prove quite useful.

Lemma 2.1. *Let $T_{n,t}$ be the distance covered in time t by a random walk on a (connected) graph with n vertices. Then $\mathbb{E}[T_{n,t}] < 1 + \sqrt{t}\sqrt{1 + 5\log n}$.*

Proof. Let $p^t(x, y)$ be the probability that a random walk that starts at vertex x will be at vertex y in exactly t steps. The Varopoulous-Carne bound [47], as formulated in [43], says

$$p^t(x, y) \leq \sqrt{e}\sqrt{\frac{\deg(y)}{\deg(x)}} \exp\left(-\frac{d(x, y)^2}{2t}\right)$$

where $d(x, y)$ is the graph distance between the two vertices. Therefore, if we consider the random walk $x_0, x_1, \ldots, x_t$ on a graph of size n and let $c \in \mathbb{R}$ be any constant, we have the following bound as a corollary of Varopoulos-Carne:

$$\begin{aligned}
\mathbb{P}(d(x_0, x_t) \geq c\sqrt{t}) &= \sum_{y:d(x_0,y)\geq c\sqrt{t}} p^t(x_0, y) \\
&\leq \sum_{y:d(x_0,y)\geq c\sqrt{t}} \sqrt{e}\sqrt{\frac{\deg(y)}{\deg(x_0)}} \exp\left(-\frac{d(x_0, y)^2}{2t}\right)
\end{aligned}$$

$$
\begin{aligned}
&< \sum_{y:d(x_0,y)\geq c\sqrt{t}} \sqrt{e}\sqrt{n}\exp\left(-\frac{c^2 t}{2t}\right) \\
&< n^{3/2}\exp\left(\frac{1-c^2}{2}\right)
\end{aligned}
$$

Letting $c = \sqrt{1+5\log n}$ therefore yields that $\mathbb{P}(d(x_0, x_t) \geq \sqrt{1+5\log n}\,\sqrt{t}) < \frac{1}{n}$.

Note that $\mathbb{E}[d(x_0, x_t)] < pn + (1-p)c\sqrt{t}$, where $p = \mathbb{P}(d(x_0, x_t) \geq c\sqrt{t})$, so we have

$$
\begin{aligned}
\mathbb{E}[d(x_0, x_t)] &\leq \frac{1}{n}n + c\sqrt{t} \\
&= 1 + \sqrt{t}\sqrt{1+5\log n}
\end{aligned}
$$

as desired. □

This bound is not tight, but it will be good enough to give us the $o(n)$ bound we seek on $\mathbb{E}[T_2]$.

Recall that D_1 is the distance between the two players at the end of Stage 1. Note that this is equivalent to the distance between the drunk's initial position and his position at the end of Stage 1. We have the following immediate corollary of Lemma 2.1.

Corollary 2.1. $\mathbb{E}[D_1] \leq 1 + \sqrt{n}\sqrt{1+5\log n}$.

Now the cop enters Stage 2. We would like to bound $\mathbb{E}[D_2]$. Note that this is equivalent to the expected distance traveled by the drunk in Stage 2.

Corollary 2.2. $\mathbb{E}[D_2] < (5\log n)^{3/4} n^{1/4}$

Proof. Using Lemma 2.1, Jensen's inequality for concave functions, and Corollary 2.1, we get

$$\begin{aligned}
\mathbb{E}[D_2] &\leq \sum_{k=0}^{n} \mathbb{P}(D_1 = k)(1 + \sqrt{k}\sqrt{1 + 5\log n}) \\
&= 1 + \sqrt{1 + 5\log n}\mathbb{E}[\sqrt{D_1}] \\
&\leq 1 + \sqrt{1 + 5\log n}\sqrt{\mathbb{E}[D_1]} \\
&\leq 1 + \sqrt{1 + 5\log n}\sqrt{1 + \sqrt{n}\sqrt{1 + 5\log n}} \\
&< (5\log n)^{3/4} n^{1/4}
\end{aligned}$$

□

Now we are done with the "gross progress" that the cop makes in Stages 1 and 2. Note that the total expected time to complete these two stages is bounded by

$$\mathbb{E}[T_1] + \mathbb{E}[T_2] \leq \text{diam}(G) + 1 + \sqrt{n}\sqrt{1 + 5\log n}.$$

2.2.3 Fine Progress

At the conclusion of stage 2, the cop's approach changes. Now she **retargets** every 4 moves. We make this notion precise in the following manner.

For each integer $j \geq 1$ let x_j, y_j be the drunk's and cop's positions, respectively, at time j (with it being the drunk's turn to move). Then in Stage 3, while $d(x_j, y_{j-1}) \geq 4$, for all j of the form $4i+1$ for some nonnegative integer i, the cop chooses x_{4i+1} as her **target** and proceeds along a geodesic toward that target for the next four steps. Consequently, the cop's target changes every 4 moves, so that for each integer $i \geq 0$, she has target x_{4i+1} at times $4i+1, 4i+2$, $4i+3$, and $4i+4$. If at time $j = 4i+1$, $d(x_j, y_{j-1}) < 4$, Stage 3 terminates and the cop's strategy moves into Stage 4, which will be described after the following lemma.

Lemma 2.2. *Let G be any graph and let $x_0 \in V(G)$ be any vertex in G. Let $\{x_0, x_1, x_2, \dots\}$ be any random walk on G beginning at x_0. Then $\mathbb{P}(d(x_0, x_4) < 4) \geq 1/s$, where $s = 4n^{2/3}$.*

Before we prove this lemma, note that we could not get away with looking at the first three steps of a random walk. That is, we could not get a bound for $\mathbb{P}(d(x_0, x_3) < 3)$ that would be useful in Equation 2.1. Consider the following example: we have a graph G with a vertex x_0. Let A_k be the set of vertices at distance k from x_0. Suppose that G looks like Figure 2.2. That is in G, $|A_1| = 1$ and $|A_2| = |A_3| = \dfrac{n-2}{2}$. Assume also that there are no edges within A_k for any k—that is,

$$|N_G(v) \cup A_k| = \emptyset$$

for all $v \in A_k$. Call any step by the random walker that guarantees $d(x_0, x_3) < 3$ a "stall." Then the probability of a stall occurring at the second step is $\dfrac{1}{(n-2)/2+1} = \dfrac{2}{n}$, and the probability of a stall occurring at the third step is $\left(1 - \dfrac{2}{n}\right)\left(\dfrac{2}{n}\right)$ since for each vertex in A_2 and A_3, there is one edge on the path toward x_0 and $\dfrac{n-2}{2}$ edges leading farther away from x_0. So then $\mathbb{P}(d(x_0, x_3) < 3) = \dfrac{2}{n} + \left(1 - \dfrac{2}{n}\right)\dfrac{2}{n} < \dfrac{4}{n}$.

We now begin the proof of Lemma 2.2.

Proof. We proceed by assuming a graph G and a vertex $x_0 \in V(G)$ exist such that there is a random walk $\{x_0, x_1, \dots\}$ with the property $\mathbb{P}(d(x_0, x_4) < 4) < 1/s$, and we shall derive a contradiction.

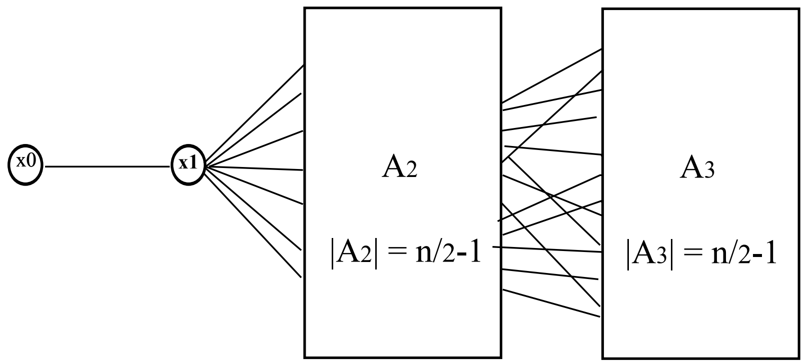

Figure 2.2: At least 4 steps are required to secure a useful bound on a random walk's progress

Let A_k be the set of vertices at distance k from x_0, and let $a_k = |A_k|$ for all k. We adapt the terms **in-degree** and **out-degree** to mean the following:

Let $v \in A_k$. Then the in-degree of v is $\deg^-(v) = |N_G(v) \cap A_{k-1}|$ and the out-degree of v is $\deg^+(v) = |N_G(v) \cap A_{k+1}|$. The corresponding term of **out-edges** of v will mean the number of edges with one end at v and the other end in A_{k+1}. We will use the notation p_G for the quantity under investigation, $\mathbb{P}(d(x_0, x_4) < 4)$, and for a vertex $v \in V(G)$, we define $p_k(v)$ to be the quantity $\mathbb{P}(d(x_0, x_4) < 4 | x_k = v)$. Note that $p_0(x_0) = p_G$ and $p_k(v) = 1$ if $v \in A_j$ for some $j < k$. Finally, we call any step by the random walker that guarantees $d(x_0, x_4) < 4$ a "stall."

We will break this proof into several statements.

Claim 2.1. *Let G' be the graph defined by removing all edges between x_0 and all but one vertex, x_1, where $p_1(x_1) = \min_{v \in A_1} p_1(v)$. Then $p_{G'} \leq p_G$.*

Proof. Since $p_G < 1/s$, there must exist a vertex $v \in A_1$ with $p_1(v) < 1/s$. Choose x_1 such that $p_1(x_1) = \min_{v \in A_1} p_1(v)$ and define G' as in the statement of the claim. Then $p_{G'} = p_1(x_1) \leq \dfrac{1}{a_1} \sum_{v \in A_1} p_1(v) = p_G$. □

Claim 2.2. *Let G'' be the induced subgraph of G' with $V(G'') = V(G') - \bigcup_{k>4} A_k$ and with all edges removed except those that are between a vertex in A_{k-1} and a vertex in A_k for $k \in \{1,2,3,4\}$. Then $p_{G''} \leq p_{G'}$.*

Proof. Let $\hat{G}'$ be the induced subgraph of G' on the vertices $V(G') - \bigcup_{k>4} A_k$ for $k > 4$. Then since $\mathbb{P}(x_t \in A_k) = 0$ when $t \leq 4$ and $k \geq 5$ (so in particular, $p_{\hat{G}'}$ and $p_{G'}$ depend only on the first four steps of a random walk originating at x_0), we have that $p_{\hat{G}'} = p_{G'}$.

Let $k \in \{1,2,3,4\}$ and let $v \in A_k$ be a vertex in $V(\hat{G}')$ with $N_{\hat{G}'}(v) \cap A_k \neq \emptyset$. If no such vertex exists then $\hat{G}' = G''$. Otherwise, let $\deg^-(v) = q, \deg^+(v) = r$, and $|N_{\hat{G}'}(v) \cap A_k| = t > 0$. $p_k(v) \geq \dfrac{q+t}{q+r+t}$. Removing the t vertices in $N_{\hat{G}'}(v) \cap A_k$ decreases $p_k(v)$ to $\dfrac{q}{q+r}$, since:

$$\begin{aligned} t > 0 \quad &\implies \quad q^2 + q + qr + tr > q^2 + q + qr \\ &\implies \quad (q+t)(q+r) > q(q+r+t) \\ &\implies \quad \frac{q+t}{q+r+t} > \frac{q}{q+r} \end{aligned}$$

Now let G'' be derived from $\hat{G}'$ by removing all edges except for those that are between A_{k-1} and A_k. (In particular, this means that for all $k \in [4]$, for all $v \in A_k \cap V(G'')$, $N_{G''}(v) \cap A_k = \emptyset$.) This decreases $p_k(v)$ for all vertices v with neighbors w such that $d(x_0, v) = d(x_0, w)$ and does not change $p_k(v)$ for all vertices v with no such neighbors. Since

$$p_{G''} \leq \frac{1}{|N_{G''}(x_1)|} \sum_{u \in N_{G''}(x_1)} \frac{1}{|N_{G''}(u)|} \sum_{v \in N_{G''}(u)} \frac{1}{|N_{G''}(v)|} \sum_{w \in N_{G''}(v)} p_4(w)$$

we have that $p_{G''} \leq p_{\hat{G}'}$. □

In view of Claims 2.1 and 2.2 above, we may assume that G has the following properties: $N_G(x_0) = x_1$, the only edges in G are between A_{k-1} and A_k for $k \in [4]$, and $A_k = \emptyset$ for all $k > 4$.

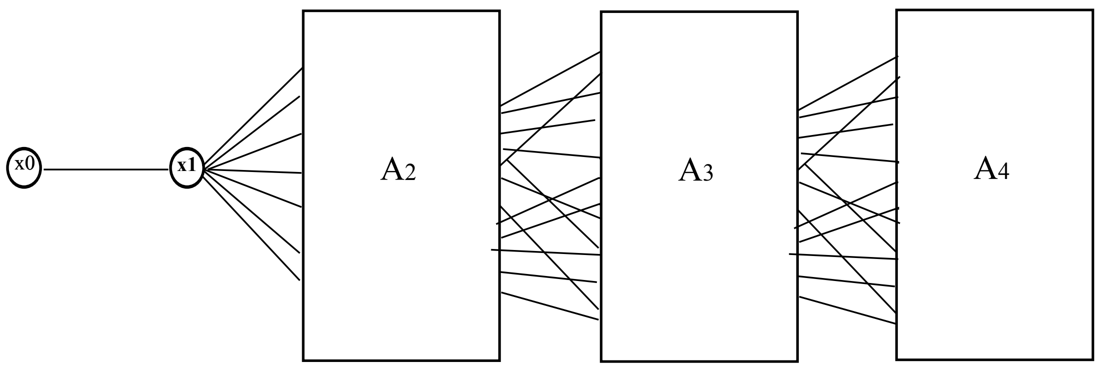

Figure 2.3: Our (alleged) counterexample G

Now define $G_k \subseteq G$ to be the induced subgraph of G on the vertices $A_k \cup A_{k+1}$ and let $e_k = |E(G_k)|$.

Claim 2.3. *$e_2 > s(s-1)$.*

Proof. We claim that the average degree of vertices in A_2 is greater than s: Let $\{d_i\}_{i=1}^{a_2}$ be the degrees of the vertices in A_2 and let $d = \sum_{i=1}^{a_2} d_i$. The vertex x_2 is chosen uniformly at random in A_2, and the probability of stalling at a vertex with degree d_i is $\frac{1}{d_i}$. Therefore the probability of stalling at A_2 is $\frac{1}{a_2}\sum_{i=1}^{a_2}\frac{1}{d_i}$. We have

$$1/s > \frac{1}{a_2}\sum_{1}^{a_2}\frac{1}{d_i} = \frac{1}{H(\{d_i\})} \geq a_2/d$$

where $H(\{d_i\})$ is the harmonic mean of the d_i. (Note that this follows from the fact that the harmonic mean is always less than or equal to the arithmetic mean.) Consequently, $d/a_2 > s$. Thus the average out-degree from A_2 is greater than $s-1$, which implies that there are more than $s(s-1)$ edges between A_2 and A_3. $\square$

Claim 2.4. *Let B be the subset of A_2 consisting of vertices with more than half of their outedges going to C, the subset of A_3 consisting of vertices with in-degree less than $n^{1/3}$. Let $b = |B|$ and $c = |C|$. Then $b < \frac{1}{2}a_2$.*

Proof. Define e_B to be the number of edges with one endpoint in B and the other in A_3. Note that $c \leq a_3 < n - a_2 \leq n - 4n^{2/3}$ and consequently the number of edges with one endpoint in A_2 and the other in C is less than $n(n^{1/3} - 4n^{2/3})$. Since more than half of the outedges of each vertex in B terminate in a vertex in C, this says that $e_B < 2n^{1/3}(n - 4n^{2/3}) = 2n^{4/3} - 8n$.

Now assume, for sake of contradiction, that $b \geq \frac{1}{2}a_2$. Then

$$\mathbb{P}(x_2 \in B) = \mathbb{P}(x_2 \in B | x_2 \in A_2)\mathbb{P}(x_2 \in A_2) \geq \frac{1}{2}\frac{s-1}{s}$$

so we have

$$\begin{aligned} 1/s &> p_G \\ &= \mathbb{P}(d(x_0, x_4) < 4 | x_2 \in B)\mathbb{P}(x_2 \in B) + \mathbb{P}(d(x_0, x_4) < 4 | x_2 \notin B)\mathbb{P}(x_2 \notin B) \\ &> \frac{s-1}{2s}\mathbb{P}(d(x_0, x_4) < 4 | x_2 \in B) \end{aligned}$$

which says that $\mathbb{P}(d(x_0, x_4) < 4 | x_2 \in B) < \dfrac{2}{s-1}$.

Let $f = \displaystyle\sum_{i=1}^{b} d_i$ where $\{d_i\}_{i=1}^{b}$ are the degrees of the vertices in B. For each vertex in B, $\mathbb{P}(d(x_0,x_4) < 4|x_2 \in B; \deg(x_2) = d_i) = \dfrac{1}{d_i}$. Since x_2 is chosen uniformly at random, we have

$$\begin{aligned}\frac{2}{s-1} > \mathbb{P}(d(x_0,x_4) < 4|x_2 \in B) &= \frac{1}{b}\sum_{i=1}^{b}\frac{1}{d_i} \\ &= \frac{1}{H(\{d_i\}_1^b)} \\ &\geq \frac{1}{(1/b)f} = \frac{b}{f}\end{aligned}$$

The average out-degree from B is $\dfrac{f}{b} - 1$ and so we get $e_B \geq b\left(\dfrac{f}{b} - 1\right) \geq \dfrac{s}{2}\left(\dfrac{s-1}{2} - 1\right) = 4n^{4/3} - 3n^{2/3}$. This is a contradiction since

$$e_B < 2n^{4/3} - 8n < 4n^{4/3} - 3n^{2/3}$$

for all integers $n \geq 1$.

Consequently, $b < \frac{1}{2}a_2$. □

Claim 2.5. *The probability that* $x_3 \in A_3 \backslash C$ *(given* $x_3 \in A_3$*) is greater than* $1/4$.

Proof. If $x_2 \in A_2$ then with probability greater than $1/2$, $x_2 \in A_2 \backslash B$. By definition, more than half of the out-edges of a vertex in $A_2 \backslash B$ terminate in $A_3 \backslash C$, and x_3 is chosen uniformly at random from the neighbors of x_2. This yields

$$\mathbb{P}(x_3 \notin C|x_3 \in A_3) = \mathbb{P}(x_3 \notin C|x_2 \in B)\mathbb{P}(x_2 \in B) + \mathbb{P}(x_3 \notin C|x_2 \notin B)\mathbb{P}(x_2 \notin B)$$

and therefore

$$\mathbb{P}(x_3 \notin C|x_3 \in A_3) \geq \mathbb{P}(x_2 \notin B|x_2 \in A_2)\mathbb{P}(x_3 \notin C|x_2 \notin B) > (1/2)(1/2) = 1/4,$$

as desired. ☐

Note that $\mathbb{P}(d(x_0, x_4) < 4 | x_3 \in A_3 \backslash C) = \dfrac{\deg^+(x_3)}{\deg(x_3)} > \dfrac{n^{1/3}}{n}$. Therefore the probability of stalling at step 3 is greater than $(1/4)\dfrac{n^{1/3}}{n} = 1/s$, yielding a contradiction. ☐

Let $j = 4i + 1$ be such that the game is in Stage 3 at time j, and let x_j, y_j be the positions of the drunk and cop, respectively, after both have moved (so that it is the drunk's turn). By Lemma 2.2 we have that $d(x_j, x_{j-4}) < 4$ with probability at least $1/s$, where $s = 4n^{2/3}$. Consequently, since the cop had x_{j-4} as her target, we now have $d(y_j, x_j) < d(y_{j-4}, x_{j-4})$ (so the distance has decreased by at least 1) with probability at least $1/s$. Let Y_i be a random variable which equals the decrease in distance between time $4(i-1)$ and $4i$. Y_i is 0 with probability less than $1 - 1/s$ and is ≥ 1 with probability at least $1/s$.

Consider the 0-1 random variable X_i with $\mathbb{P}(X_i = 1) = 1/s$ for all i (note $\mathbb{E}[X_i] \leq \mathbb{E}[Y_i]$ for all i). Let $S_n = X_1 + \cdots + X_n$, for all $n \in \mathbb{N}$. Consider the random process $\{X_i : i \in \mathbb{N}\}$ with the stopping rule that says the process terminates at time τ if $S_\tau = D_2 - 3$. By Wald's identity [48], $\mathbb{E}[S_\tau] = \mathbb{E}[\tau]\mathbb{E}[X_i]$. Since $\mathbb{E}[S_\tau] = \mathbb{E}[D_2] - 3$, we have that the expected stopping time $\mathbb{E}[\tau] = \dfrac{\mathbb{E}[D_2] - 3}{1/s}$. This is the expected number of retargetings needed to get $S_\tau = D_2 - 3$, so we have

$$\mathbb{E}[T_3] \leq 4\mathbb{E}[\tau] = 4s(\mathbb{E}[D_2] - 3) < 4((5 \log n)^{3/4} n^{1/4} - 3)(4n^{2/3})$$

Stage 3 terminates when the distance between the cop and the drunk is less than four, and it is the cop's turn. In Stage 4, which terminates when the drunk is captured, the cop uses the greedy strategy, defined as follows. Suppose that the strategy enters Stage 4 at time t, during which time the drunk is at vertex x_t and the cop is about to move from vertex y_{t-1}. Then $d(x_t, y_{t-1}) \leq 3$, and the cop moves such that $d(x_t, y_t) \leq 2$. Now for any $r > t$, if the drunk moves such that $d(x_r, y_{r-1}) = 3$, the cop can choose y_r to ensure that $d(x_r, y_r) = 2$. For each r, with probability at least $1/\Delta$, the drunk moves "toward" the cop—i.e., such that $d(x_r, y_{r-1}) = 1$; if that happens, the cop can choose $y_r = x_r$, capturing the drunk. This takes at most Δ expected moves, so $\mathbb{E}[T_4] \leq \Delta$ where T_4 is the expected time spent in Stage 4.

Adding together our results about the expected time to complete each of the four stages yields the following bound on the expected capture time:

$$\begin{aligned}\sum_{i=1}^{4} \mathbb{E}[T_i] &\leq \text{diam}(G) + \mathbb{E}[D_1] + 16n^{2/3}(\mathbb{E}[D_2] - 3) + \Delta \\ &< \text{diam}(G) + 1 + \sqrt{n(1 + 5\log n)} + 4((5\log n)^{3/4} n^{1/4} - 3)(4n^{2/3}) + \Delta \\ &= \text{diam}(G) + \Delta + o(n)\end{aligned}$$

In fact, we can bound diam(G)+Δ with a bit of graph theory.

Lemma 2.3. *For any graph G with $|V(G)|$=n, diam$(G) + \Delta \leq n + 1$.*

Proof. Assume, for sake of contradiction, that there is a graph G such that diam$(G) > n - \Delta + 1$. Let $u, v, w \in V(G)$ be (not necessarily distinct) vertices in G such that $\deg(v) = \Delta$ and $d(u, w) = d \geq n - \Delta + 2$. Now we break this proof into two cases:

Case 1: v lies on a shortest path between u and w.

Let P_1 be a shortest $u - - - w$ path containing v. At most two neighbors of v may lie on P_1, so there are at least $\Delta - 2$ vertices not on P_1. Since the length of P_1 is at least $\geq n - \Delta + 2$, there are at least $n - \Delta + 3$ vertices in P_1. But now we have that $|V(G)| \geq \Delta - 2 + n - \Delta + 3 > n$, which is a contradiction.

Case 2: v is not on any shortest $u - - - w$ path.

Let P_2 be a shortest $u---w$ path. If more than 2 neighbors of v are in P_2, then v is also on a shortest $u---w$ path (let x_1, x_2, and x_3 be the neighbors of v on P_2, appearing in that order; then the section involving the three neighbors of v could be replaced with $x_1 - v - x_3$ to create another shortest $u---w$ path). Therefore there are at least $\Delta - 1$ vertices not on P_2 (v and $\Delta - 2$ of its neighbors), and at least $n - \Delta + 3$ vertices on this path. So once again, $|V(G)| \geq \Delta - 1 + n - \Delta + 3 > n$, which is a contradiction.

Therefore $\text{diam}(G) \leq n - \Delta + 1$ for all graphs G. □

Therefore we have the following theorem about the expected capture time.

Theorem 2.1. *On a connected, undirected, simple graph on n vertices, a cop with the described four-stage strategy will capture a drunk in expected time $n+o(n)$.*

2.3 Generalizations and Variations

The reader may, for instance, have noticed that in Example (4) in Section 2.1, we considered a cop who was not only greedy but also rather insistently foolish. What about the greedy cop who makes distance-minimizing decisions at random? The "ladder to the basement" graph is no longer a problem for her, (the expected capture time in this example is now less than n). Is it possible that the greedy algorithm with disputes settled by a random decision between choices is enough to guarantee time $n+o(n)$?

It is also possible that a deterministic greedy cop who breaks ties by considering her distance to vertices previously occupied by the drunk will capture in expected time at most $n+o(n)$.

2.3 Generalizations and Variations

An alternative greedy strategy, suggested by Andrew Beveridge [9], concerns itself with minimizing the drunk's expected hitting time to the cop at every step. It would be interesting to see if this strategy also has expected capture time at most $n + o(n)$.

Another problem to consider is that of the "invisible" drunk. That is, we are once again in the cop vs. drunk situation, but this time, the cop has no information about her opponent's whereabouts until she captures him, again by simultaneously occupying the same vertex. For more thoughts on this variation, see Chapter 7.

Chapter 3

Cops in the Dark: Depth First Pursuit

In this chapter, we consider a version of Cops and Robbers played in the dark—that is, the players are invisible to each other. We consider a particular strategy for the cop: depth first pursuit. We investigate how well this strategy performs against invisible adversaries in several different environments.

3.1 Cops & Sitters

In this section we discuss a variant of the game in which the cop moves as originally designed, but the robber—now a "sitter" or "immobile hider"—chooses a vertex of the graph and stays there until the cop captures him. The cop's initial position is known to the sitter (which we encode by selecting a vertex of the graph to be the "root"). The players do not know each other's strategies (in particular, the cop does not know the sitter's position, or this game would be over quicker than the sitter might like). The sitter chooses his hiding vertex according to his strategy (which is a probability distribution on the vertex set) and the cop performs a walk beginning at the root according to her strategy (which is a probability distribution on walks that begin at the root). Note that *a priori*, one or both of these strategies may be deterministic (i.e. a probability distribution that gives probability 1 to one option and 0 to every other). This variation was discussed in [23] on a network, where the author notes that the value of the game ν is $|E(G)|$ on a tree, $|E(G)|/2$ on an Eulerian graph G, and in general satisfies

$$\frac{1}{2}\mu \leq \nu \leq \frac{1}{2}L(S^*)$$

where μ is the length of the network (in our case, this is the number of edges in our graph) and $L(S^*)$ is the minimum length of a closed curve passing through all of the points of the network (on a graph, this is the minimum length of a walk that begins and ends at the same vertex and visits every vertex in the graph at least once). Note that $L(S^*) \leq 2\mu$ for any network and so the value of the game on a graph is between $|E(G)|/2$ and $|E(G)|$. We define and analyze the depth first pursuit strategy, and find that it is an optimal strategy on a tree. We also find that this strategy yields an expected capture time between $\frac{n+1}{2}$ and $n-1$ for all graphs (strictly less than n for any graph containing a cycle).

3.1.1 Player Strategies

In order to define the cop strategy—a uniformly random depth first pursuit—we develop some machinery in order to be able to generate such a strategy easily. We begin by defining the strategy on a tree.

Definition 3.1. Given a rooted tree G with root r, we define **the uniform algorithm** (henceforth referred to as **uni-alg**) in the following manner. Begin with $\sigma=\{r\}$, $v=r$ and $U=\{v \in V(G) : \deg(v) = 1\}$ (the set of leaves of G). Visitation schedule: Select a previously unvisited child u of v uniformly at random (if it exists) and append this to list σ. If no unvisited children of v remain, then select u to be the parent node of v. Set $v=u$. If $u \in U$, set $U=U\backslash\{u\}$. Repeat visitation schedule, terminating when $U = \emptyset$. Uni-alg outputs the vertex sequence σ.

Note that all of the vertices in $V(G)$ appear at least once in σ for any σ that results from an instance of uni-alg.

Definition 3.2. A walk on the vertices of a tree G with root r which starts at r is a **DFP** (depth first pursuit) if whenever the walk enters a branch B defined by a vertex v (i.e. v and all of its descendants), it does not exit B until all $w \in B$ have been visited at least once.

Claim 3.1. *A DFP on G is uniquely determined by an ordering on the leaves of G.*

Proof. Note that this is not a one-to-one correspondence, as not all leaf orderings result in a valid DFP. However, given a valid ordering $\ell = \{\ell_1, \ell_2, \ldots, \ell_k\}$ on the leaves of G (i.e. a leaf ordering that results from a DFP), we can define a DFP by taking the (unique) shortest path from r to ℓ_1, concatenating it with the shortest path from ℓ_1 to ℓ_2, and so on. Note that taking two different leaf orderings ℓ and ω would yield two different walks on the vertices via this process. Consequently, a DFP is uniquely determined by a leaf ordering. □

Claim 3.2. *Uni-alg generates a unique DFP.*

Proof. First we must show that uni-alg generates a DFP. According to uni-alg, once a vertex $v \in V(G)$ is reached, its parent is not visited again until all of its children are visited at least once. Therefore, all of the descendants of v are visited before its parent is visited again. As this is true for all vertices $v \in V(G)$, the sequence generated by uni-alg is a DFP.

Now suppose that σ and ρ are two different sequences generated by uni-alg. Let σ_ℓ and ρ_ℓ be the orders in which the leaves of G are visited by σ and ρ, respectively. By Claim 3.1, these two leaf orderings correspond to the same DFP if and only if $\rho_\ell = \sigma_\ell$. Suppose that this equality holds and let $\sigma_\ell = \rho_\ell = \{\ell_1, \ell_2, \ldots, \ell_k\}$. By the construction of uni-alg, σ and ρ must both take the (unique) shortest path from r to ℓ_1, then from ℓ_1 to ℓ_2, and so on, and so $\sigma = \rho$.

Therefore, uni-alg generates a unique DFP. □

Claim 3.3. *Let σ be a DFP. There is a unique outcome of uni-alg that yields σ.*

Proof. We will first show that σ is a possible outcome of uni-alg. Suppose that the DFP $\sigma = \{v_1{=}r, v_2, \ldots, v_M\}$ with $\{\ell_1, \ell_2, \ldots, \ell_k\}$ the subsequence that gives the order of leaves hit by σ. Then we know that σ consists of the shortest r—ℓ_1 path concatenated with the shortest ℓ_1—ℓ_2 path and so on. This DFP is clearly yielded by the outcome of uni-alg that selects children in the corresponding order.

Now suppose that σ and ρ are two different DFP strategies. By Claim 3.1, the respective leaf orderings σ_ℓ and ρ_ℓ are different. But as in the proof of Claim 3.2, outcomes of uni-alg are uniquely determined by leaf orderings, and so σ and ρ must be generated by two different outcomes of uni-alg. □

Claims 3.2 and 3.3 yield Lemma 3.1.

Lemma 3.1. *There is a one-to-one correspondence between the outcomes of uni-alg and the set of DFP on a given tree.*

Now we are ready to prove the following lemma.

Lemma 3.2. *Uni-alg results in a uniformly random DFP.*

Proof. Let $\sigma = \{v_1, v_2, \ldots, v_M\}$ be a sequence given by uni-alg (which by Lemma 3.1 makes it a DFP). For all $v \in V(G)$ and $t \in \mathbb{N}$, define $c(v,t)$ to be the number of children of v that have not yet been visited by time t wherever this quantity is positive. Then at each time t in uni-alg, the probability of selecting a given child of v_t was $\dfrac{1}{c(v,t)}$. Where no such child exists, a non-random path was followed until a time t where a vertex v_t was reached with $c(v,t) > 0$. Recall that σ visited all vertices of $V(G)$, so

$$\mathbb{P}(\sigma) = \prod_{v_i \in \sigma : c(v,i) \neq 0} \frac{1}{c(v_i)}$$

Each non-leaf vertex $v \in V(G)$ contributes $\dfrac{1}{c(v)}$ to this count (where $c(v)$ is the number of children of v): suppose that v is visited at times $t_1, t_2, \ldots, t_k$; then $c(v, t_1) = c(v)$, $c(v, t_{k-1}) = 1$ and for all i, $c(v, t_{i+1}) = c(v, t_i) - 1$. Therefore we have

$$\mathbb{P}(\sigma) = \prod_{v \in V(G) : \deg(v) > 1} \frac{1}{c(v)!}$$

Since this value does not depend on σ, any sequence that can result from uni-alg (which by Lemma 3.1 is any DFP) is equally likely. □

We first consider the game played against an immobile hider (sitter) on a tree.

Lemma 3.3. *A uniformly random DFP strategy against the sitter on a tree with m edges gives expected capture time $T_m \leq m$.*

Proof. We proceed by induction on m. When $m = 1$, the problem is trivial: the sitter will either choose the root or not, with probabilities p and $q = 1 - p$ respectively, for an expected capture time of $T_1 = q \leq 1$. Suppose now that for all trees with fewer than m edges, the capture time is at most the number of edges, and let G be a tree with m edges.

Suppose that $B_1, B_2, \ldots, B_k$ are the main branches of G, and for each $i \in \{1..k\}$ let p_i be the probability that the sitter is in branch B_i. Since the cop is performing a uniformly random DFP (which by Claim 3.2 is equivalent to performing uni-alg), she has probability $\frac{1}{k}$ of first choosing branch B_i for all $i \in [k]$. For each i let $|E(B_i)| = s_i$.

Note that if she chooses B_i as her first branch, then with probability p_i the sitter is there, and by the induction hypothesis she finds him in time at most s_i. Otherwise, she takes time $2s_i$ to complete a DFP in branch B_i before returning to the root, and by the induction hypothesis, she finds the sitter in $G - B_i$ in time at most $m - s_i$. Thus, we have that the expected capture time T_m satisfies

$$\begin{aligned}
T_m &\leq \frac{1}{k}\sum_{i=1}^{k}(p_i s_i + (1 - p_i)(m + s_i)) \\
&= \frac{1}{k}\sum_{i=1}^{k}((1 - p_i)m + s_i) \\
&= \frac{1}{k}\left(\sum_{i=1}^{k} m - m\sum_{i=1}^{k} p_i + \sum_{i=1}^{k} s_i\right) \\
&= \frac{1}{k}(mk - m + m) \\
&= m
\end{aligned}$$

By induction, this yields our desired result. □

We will proceed to show that the value of the searcher/pursuer (cop) vs. immobile hider (sitter) game on a tree with m edges is in fact m, and in particular, that this value is achieved by a cop using the DFP search strategy. We will also find the unique optimal strategy for the sitter. We define this strategy now.

In what follows, $\mathcal{B}(G)$ refers to the set of branches of G. A branch B_v of G is said to be defined by v if it consists of v and all of its descendants. When considered as a tree, B_v is a rooted tree with root v. $\mathcal{B}_{\mathcal{G}}(v)$ will denote the set of main branches of B_v.

Consider a walk X on G which begins at r, whose behavior is defined in the following lemmas, and which terminates when it hits a leaf for the first time.

Definition 3.3. For all $B \in \mathcal{B}_{\mathcal{G}}(v)$, let

$$\omega_G(B|B_v) = \mathbb{P}(X \text{ terminates in } B|X \text{ terminates in } B_v) = \frac{|E(B)|}{\sum\limits_{C \in \mathcal{B}_{\mathcal{G}}(v)} (1 + |E(C)|)}$$

Furthermore, for any $B \subseteq C$, if X is a walk on C, let

$$\omega_C(B) = \mathbb{P}(X \text{ terminates in } B)$$

Note that w_G satisfies the following condition.

Lemma 3.4. *Let $B_w \subseteq B_v \subseteq G$ be branches of G defined by w and v, respectively. Then $\omega_G(B_w|B_v) = \omega_{B_v}(B_w)$.*

Proof. By definition,

$$\omega_G(B_w|B_v) = \frac{|E(B_w)|}{\sum_{C \in \mathcal{B}_{\mathcal{G}}(v)}(1 + |E(C)|)}.$$

Note that $|E(B_v)| = \sum_{C \in \mathcal{B}_{\mathcal{G}}(v)} (1 + |E(C)|)$. Consequently,

$$\omega_G(B_w|B_v) = \frac{|E(B_w)|}{|E(B_v)|} = \omega_{B_v}(B_w).$$

□

Lemma 3.5. *Let $\mathcal{L}(G) = \{B_v \subseteq G | \deg(v) = 1\}$. (So B_v is a singleton for all such v, containing only the leaf v.) Then w_G is a probability distribution on $\mathcal{L}(G)$.*

Proof. $\sum_{L \in \mathcal{L}(G)} w_G(L) = \mathbb{P}(X \text{ terminates in } \mathcal{L}(G)) = 1$, since X terminates at a leaf and the set of leaves is contained in the set of branches in $\mathcal{L}(G)$. □

Let $L(G) \subseteq V(G)$ be the set of leaves of G and define $p_G : L(G) \to [0,1]$ to be $p_G(\ell) = \omega_G(B_\ell)$. By Lemma 3.5, p_G is a probability distribution on the leaves of G.

Now we are ready to define the strategy for the sitter.

Definition 3.4. According to the **sitter strategy** $\mathcal{S}$, the sitter chooses any leaf ℓ with probability $p_G(\ell)$, and any non-leaf vertex v with probability 0.

3.1.2 Value of the Cop vs. Sitter Game

Lemma 3.6. *In the cop vs. sitter game played on any tree with m edges, the expected capture time, T_m, is at least m.*

Proof. Let G be a tree with root r. Suppose that G has main branches $B_1, B_2, \ldots, B_b$. For each $i \in \{1..b\}$, let $s_i = |E(B_i)|$. The cop starts at r and performs some strategy $\mathcal{P}$, according to which the probability that the cop chooses branch B_i first is p_i. Note that according to the sitter strategy defined in Definition 3.4, the probability that the sitter is in B_i is $\frac{s_i}{m}$.

We proceed by induction on m. When $m = 1$ we have a rooted tree on two vertices. With probability 1 the sitter chooses the non-root vertex, so the expected capture time is $T_1 = 1$. Now suppose that $T_k \geq k$ for all $k < m$ and suppose G has m vertices. Since p_i, s_i, and m are all nonnegative, we have that

$$
\begin{aligned}
0 \leq \sum_{i=1}^{b} p_i \frac{s_i^2}{m} \quad &\implies \quad 0 \geq -\sum_{i=1}^{b} p_i \frac{s_i^2}{m} \\
&\implies \quad \sum_{i=1}^{b} p_i s_i \geq \sum_{i=1}^{b} p_i \left(s_i - \frac{s_i^2}{m} \right) \\
&\implies \quad \sum_{i=1}^{b} p_i s_i \geq \sum_{i=1}^{b} p_i \frac{s_i}{m}(m - s_i)
\end{aligned}
$$

Note that the probability C_i that the cop captures the sitter in B_i is at most $\frac{s_i}{m}$ (since the sitter is only in B_i with probability s_i/m). By the induction hypothesis, if the sitter is captured in branch B_i, the expected time spent in this branch is at least s_i. Therefore we will suppose that the cop spends at least s_i turns in B_i before exiting it (since by the induction hypothesis, if she spends less than s_i turns in branch B_i, she could not have captured the sitter there). Thus,

$$
\begin{aligned}
\sum_{i=1}^{b} p_i s_i \geq \sum_{i=1}^{b} p_i \frac{s_i}{m}(m - s_i) \quad &\implies \quad \sum_{i=1}^{b} p_i s_i \geq \sum_{i=1}^{b} p_i C_i (m - s_i) \\
&\implies \quad \sum_{i=1}^{b} p_i s_i - \sum_{i=1}^{b} p_i C_i (m - s_i) + m \geq m \\
&\implies \quad \sum_{i=1}^{b} p_i s_i - \sum_{i=1}^{b} p_i C_i (m - s_i) + \sum_{i=1}^{b} p_i m \geq m
\end{aligned}
$$

$$\implies \sum_{i=1}^{b} p_i \left[s_i + m - C_i(m - s_i) \right] \geq m$$

$$\implies \sum_{i=1}^{b} p_i \left[C_i s_i + (1 - C_i) \left(m + \frac{s_i}{1 - C_i} \right) \right] \geq m$$

By the induction hypothesis we have that if the sitter is caught in the first branch that the cop visits (B_i, say), then this is done in no fewer than s_i moves. Furthermore, if the sitter is not caught in the B_i, and supposing that the cop spent time $2t_i$ in B_i before giving up (note that she must spend an even amount of time since she is on a tree and begins and ends at the same point), then applying the induction hypothesis to $G \backslash B_i$ gives that the cop spent at least $2t_i + m - s_i$ moves in $G \backslash B_i$. Therefore we have

$$T_m \geq \sum_{i=1}^{b} p_i [C_i s_i + (1 - C_i)(2t_i + m - s_i)]$$

where $C_i = \dfrac{s_i}{m} \dfrac{t_i}{s_i} = \dfrac{t_i}{m}$.

Case 1: $t_i < s_i$. That is, the cop does not visit all of the edges (and consequently vertices) in B_i before leaving the branch.

We know that

$$\sum_{i=1}^{b} p_i \left[C_i s_i + (1 - C_i) \left(m + \frac{s_i}{1 - C_i} \right) \right] \geq m$$

and we would like to show that

$$\sum_{i=1}^{b} p_i [C_i s_i + (1 - C_i)(2t_i + m - s_i)] \geq m$$

so it suffices to show that

$$\sum_{i=1}^{b} p_i \left[C_i s_i + (1 - C_i) \left(m + \frac{s_i}{1 - C_i}\right)\right] \leq \sum_{i=1}^{b} p_i[C_i s_i + (1 - C_i)(2t_i + m - s_i)].$$

By the induction hypothesis, $2t_i \geq s_i$. Therefore,

$$\begin{aligned} 2t_i - s_i \geq 0 \quad &\implies \quad s_i - (1 - t_i/m)(2t_i - s_i) \geq 0 \\ &\implies \quad \sum_{i=1}^{b} p_i(s_i - (1 - t_i/m)(2t_i - s_i)) \geq 0 \\ &\implies \quad \sum_{i=1}^{b} p_i s_i \geq \sum_{i=1}^{b} p_i(1 - t_i/m)(2t_i - s_i) \end{aligned}$$

Therefore,

$$\sum_{i=1}^{b} p_i(s_i + s_i t_i/m + (1 - t_i/m)m) \geq \sum_{i=1}^{b} p_i(s_i t_i/m + (1 - t_i/m)(2t_i + m - s_i))$$

as desired.

Case 2: $t_i = s_i$.

Then we have that the expected capture time T_m satisfies

$$\begin{aligned} T_m \quad &\geq \quad \sum_{i=}^{b} p_i(s_i^2/m + (1 - s_i/m)(2s_i + m - s_i)) \\ &= \quad \sum_{i=1}^{b} p_i(s_i^2/m + (1 - s_i/m)(m + s_i)) \\ &= \quad \sum_{i=1}^{b} p_i m = m \end{aligned}$$

Therefore, the expected capture time is at least m for all trees with m

edges.

□

From Lemmas 3.3 and 3.6, we immediately get the following lemma.

Lemma 3.7. *The DFP strategy defined by uni-alg in Definition 3.1 is an optimal strategy for the cop on a tree. The sitter strategy $\mathcal{S}$ defined in Definition 3.4 is an optimal strategy for the sitter on a tree.*

We also have our desired theorem.

Theorem 3.1. *The cop vs. sitter game played on a rooted tree with m edges has value m.*

Note that not only do we know that that $\mathcal{S}$ is an optimal strategy for the sitter, but also that no other strategy would do. We show this in the proof of the following lemma.

Lemma 3.8. *The sitter strategy $\mathcal{S}$ is the unique optimal strategy for the sitter on a rooted tree with m edges.*

Proof. We proceed to prove the statement, "the cop can capture any sitter not using strategy $\mathcal{S}$ in fewer than m expected turns" by induction on m. When m=1 we have a rooted copy of P_2, so any sitter strategy which differs from $\mathcal{S}$ must involve a non-zero probability of going to the root and therefore being captured in time 0. Since the strategy which places the sitter at the non-root vertex with probability 1 has expected capture time 1, this strategy must have expected capture time less than 1 (specifically $1-q$ where $q > 0$ is the probability that the sitter is at the root).

Now suppose that the statement is true for all $r < m$, and consider a tree G with m edges.

Let $\sigma \neq \mathcal{S}$ be a different strategy for the sitter on G. Suppose G has main branches $B_1, \ldots, B_k$, with corresponding sizes $s_1, \ldots, s_k$. Then there exists some branch B_i where the probability p that the sitter is in branch B_i is not equal to s_i/m. So without loss of generality, let B_i be chosen such that $p > s_i/m$. Consider the cop strategy in which she visits B_i first. With probability p, she finds the sitter there, and by the induction hypothesis, this takes time at most s_i. With probability $1-p$, she spends time $2s_i$ in B_i and then finds the sitter in $G \backslash B_i$, which by the induction hypothesis takes time at most $m-s_i$. Then we have that the expected capture time satisfies the inequality

$$T < ps_i + (1-p)(2s_i + m - s_i) = ps_i + (1-p)(m + s_i) = s_i + (1-p)m$$

Recall that we assumed $p > s_i/m$, so we have

$$\begin{aligned} p > s_i/m \quad &\Longrightarrow \quad s_i < pm \\ &\Longrightarrow \quad s_i < m - (1-p)m \\ &\Longrightarrow \quad s_i + (1-p)m < m \end{aligned}$$

Consequently, we know that the cop can capture any sitter not using strategy $\mathcal{S}$ in fewer than m expected turns, and therefore $\mathcal{S}$ is the unique optimal strategy on G. □

Theorem 3.2. *The cop vs. sitter game on any graph G of size n takes no more than $n-1$ expected moves, with equality holding if and only if G is a tree.*

In light of Theorem 3.1, it suffices to show that the expected capture time on any graph G containing at least one cycle is strictly less than n. To prove this, it will be handy to have the following fact.

Lemma 3.9. *For any tree T, let $\mathcal{S}(T)$ be the (unique, by Lemma 3.8) optimal sitter strategy $\mathcal{S}$ on T. Let G be any graph on n vertices. Then there exist two spanning trees $T_1 \neq T_2$ of G such that $\mathcal{S}(T_1) \neq \mathcal{S}(T_2)$.*

Proof. Suppose first that $G{=}C_n$, the (rooted) cycle on n vertices, with vertices $v_1, v_2, \ldots, v_n$. Without loss of generality, suppose that v_1 is the root. Then $\mathcal{S}(T_1)$ on the spanning tree $T_1 = G\backslash\{v_1v_n\}$ gives probability 1 to v_n, whereas $\mathcal{S}(T_2)$ on the spanning tree $T_2 = G\backslash\{v_{n-1}v_n\}$ gives probability $1/(n-2)$ to v_n and $(n-3)/(n-2)$ to v_{n-1}. So we have two spanning trees T_1 and T_2 that satisfy $S(T_1){\neq}S(T_2)$.

Otherwise, if G is a rooted, connected, non-tree graph such that $G{\neq}C_n$, we observe first that we can remove a vertex $v \in V(G)$ such that $G\backslash\{v\}$ is a connected graph which is not a tree (since if G contains two cycles, we can remove a vertex on one cycle without removing the other cycle; otherwise G contains a cycle with some trees appended to it, in which case we can remove a leaf from one of those trees). Now we proceed by induction on n to prove that for any such G on n vertices, there are two spanning trees T_1 and T_2 such that the number of leaves in T_1 is less than the number of leaves in T_2. When $n{=}3$ we have that $G{=}K_3$ with vertices v_1, v_2, v_3 where without loss of generality, v_1 is the root. Then we have the spanning trees $T_1 = \{v_1v_2, v_2v_3\}$, which has only v_3 as a leaf, and $T_2 = \{v_1v_2, v_1v_3\}$, which has both v_2 and v_3 as leaves.

Now suppose that the statement is true for any graph on $n{-}1$ vertices satisfying the hypotheses above, and let G be such a graph on n vertices. Remove any vertex such that $G\backslash\{v\}$ is still connected and contains a cycle (which is possible by the observation in the paragraph above). By the induction hypothesis, $G\backslash\{v\}$ has two spanning trees with a distinct number of leaves. Appending v to these two spanning trees therefore yields two spanning trees of G with a distinct number of leaves (since v is a leaf in both cases).

Therefore we have shown that any rooted, connected, cyclic graph G has at least two spanning trees T_1 and T_2 with a different number of leaves, and therefore with $\mathcal{S}(T_1) \neq \mathcal{S}(T_2)$. □

Now, with the help of Lemma 3.9, we proceed with the proof of Theorem 3.2.

Proof. Let G be a rooted, connected graph containing at least one cycle. Then by Lemma 3.9 we can choose two spanning trees T_1, T_2 with $\mathcal{S}(T_1) \neq \mathcal{S}(T_2)$. Recall that by Lemma 3.8 we have that these are the unique sitter strategies on T_1 and T_2, respectively, which give expected capture time at least m. So if the sitter chooses a strategy that differs from $\mathcal{S}(T_1)$, then the cop who plays her uniform DFP strategy on T_1 will get expected capture time less than $n-1$. Similarly, if the sitter chooses $\mathcal{S}(T_1)$ as his strategy, the cop who plays the DFP strategy on T_2 will get expected capture time less than $n-1$. Therefore, the cop who chooses a uniformly random spanning tree and performs DFP on that tree will get expected capture time less than $n-1$. □

We can also give a lower bound on the expected capture time in this game (in any graph).

Lemma 3.10. *The expected capture time T satisfies $T \geq \dfrac{n+1}{2}$ for all graphs G on n vertices.*

Proof. First we consider this game played on K_n. Note that all cop strategies which consist of a permutation of the vertices are equivalent, and give expected capture time at least $\dfrac{n+1}{2}$: the cop has probability $\frac{1}{n}$ of finding the sitter on step 1, $(1-\frac{1}{n})\frac{1}{n-1} = \frac{1}{n}$

of missing the sitter on step 1 and finding him on step 2, and so on, for a total expected capture time of $\frac{1}{n}(1+2+\cdots+n)=\frac{n+1}{2}$. Any cop strategy which repeats a vertex would give a higher expected capture time, since repetition of a vertex does not contribute to finding an immobile hider.

The case of K_n is the "easiest" for the cop, in the following sense. We note that since the value (we will refer to it now as $ECT(G)$) of this cop vs. sitter game on a graph G exists for any (connected) G, and we have

$$\sup_{\tau}\inf_{\sigma} ECT^{\sigma}_{\tau}(G) \leq ECT(G) \leq \inf_{\sigma}\sup_{\tau} ECT^{\sigma}_{\tau}(G),$$

where $ECT^{\sigma}_{\tau}(G)$ is the expected capture time when the cop plays strategy σ and the sitter plays strategy τ, and the supremums are taken over all robber strategies τ and the infimums are taken over all cop strategies σ, we have that both of these inequalities are actually equalities.

Therefore, we can show that K_n is the easiest graph for the cop by proving that adding an edge to a graph G cannot increase the expected capture time:

Let $G^* = G \cup \{uv\}$ for some $u, v \in V(G)$, and let Σ_G be the set of possible cop strategies on G, so $\Sigma_G \subseteq \Sigma_{G^*}$. Therefore

$$ECT(G^*) = \sup_{\tau}\inf_{\sigma\in\Sigma_{G^*}} ECT^{\sigma}_{\tau}(G^*) \leq \sup_{\tau}\inf_{\sigma\in\Sigma_{G}} ECT^{\sigma}_{\tau}(G) = ECT(G),$$

as desired. □

We have shown that the depth first pursuit strategy is optimal for the cop chasing an immobile adverary. In the next section, we consider how well this strategy behaves on K_n against an invisible, mobile robber.

3.2 Cops & Robbers on K_n

We continue in this section to consider the cop vs. robber game played in the dark with simultaneous moves, but now the robber is not restricted to the immobile hider strategy in Section 3.1. We investigate this game played on the complete graph K_n with vertices labeled with the numbers from $\{1..n\} = \{1, 2, 3, \ldots, n\}$. Both players are intelligent, with the robber attempting to maximize the expected length of the game and the cop trying to minimize it, as usual.

It is not difficult to see that the value of this game is n:

Claim 1 *The robber can guarantee that the expected capture time is at least n on this graph.*

Proof. Consider the robber who goes to each vertex with probability $1/n$ at each time t. The cop captures him at each time with probability $1/n$, so an average of n steps are required to capture. □

Claim 2 *The cop can guarantee that the expected capture time is at most n on this graph.*

Proof. The cop performing the same strategy as the robber described in Claim 1 (going to a given vertex with probability $1/n$ at each step) has probability $1/n$ of capturing the robber at every step, and so captures in n steps on average. □

We would like to see how close to this value the cop can get by using uniformly random depth first pursuit (see Definition 3.2). We proceed by considering the random depth first pursuit strategy for the cop, which in this version of events is equivalent to choosing a permutation of the labels in $\{1..n\}$ uniformly at random; if she fails to catch the robber in one pass through the corresponding element of S_n, she next chooses a uniformly random permutation in S_n (independently of her history of choices). We will show that an optimal strategy for the robber against DFP is to perform a random permutation of the vertices as well, and we give a bound on the capture time. We also show that DFP is asymptotically optimal for the cop.

3.2.1 Expected Capture Time

Let W_n be the set of length n words in alphabet $\{1..n\}$. Let $\sigma \in W_n$ be such that there exists $r \in \{1..n\}$ where r appears more than once in σ, with the last appearance occurring in position k. Then there is some $s \in \{1..n\}$ that does not appear in σ. Define the sequence $\tau \in W_n$ to be the same as σ except in position k, in which τ has an s instead of an r.

For any sequence $\nu \in W_n$ define $\text{Hit}(\nu)$ to be the set of permutations γ in S_n such that for some $j \in \{1..n\}$, ν and γ agree in the j^{th} position. Consider the map φ such that for all $\gamma \in S_n$, $\varphi(\gamma) = (rs)\gamma$ (that is, it switches the positions of the r and the s in γ). The following lemmas will be used to help show that there are more permutations (equally likely cop strategies) that hit σ than that hit τ, and consequently that permutations are a better choice for the robber's strategy than sequences with repetitions.

Lemma 3.11. *φ is an injection from $A = Hit(\tau)\backslash Hit(\sigma)$ into $B = Hit(\sigma)\backslash Hit(\tau)$.*

Proof. First we show that φ is map A to B. Let $\rho \in A$, so ρ intersects with τ only at the k^{th} position, where they both have an s. But we know that in the k^{th} position of ρ is an r and therefore in the k^{th} position of $\varphi(\rho)$ is an s, and so $\varphi(\rho)$ hits σ but not τ.

Now suppose that $\alpha, \beta \in A$ are two different permutations. Then clearly

$$\varphi(\alpha) = (rs)\alpha \neq (rs)\beta = \varphi(\beta).$$

Therefore φ is an injection from A into B. □

Lemma 3.12. *φ is not a surjection from A to B.*

Proof. Let $\gamma \in B$ be a permutation that has an s in the i^{th} position (where i is one of the positions in which σ has an r) and an r in the k^{th} position. Suppose, for contradiction, that there exists $\rho \in A$ such that $\gamma = \varphi(\rho)$. Note that $\varphi(\gamma)$ has an r in the i^{th} position and an s in the k^{th} position; in all other positions, $\varphi(\gamma)$ misses both σ and τ, but it hits σ at position i and it hits τ in positions i and j. Consequently $\varphi(\gamma) \notin A$, but $\varphi(\gamma) = \varphi(\varphi(\rho)) = \rho \in A$, which is a contradiction. Consequently φ is not a surjection. □

As a consequence of Lemmas 3.11 and 3.12, we have the following fact.

Theorem 3.3. *Let p_ω be the probability that the robber escapes after n moves using strategy ω. Then p_ω is maximized when ω is a permutation.*

Proof. By Lemmas 3.11 and 3.12, there are more permutations that intersect with σ than that intersect with τ. We show that there are more permutations that intersect with a sequence with repetitions than that intersect with a permutation:

Note that by symmetry, p_ω is the same for any permutation ω. We will call this common value p. Suppose that for some word θ, $p_\theta > p$ and that p_θ is maximal over all words. Clearly θ is not a permutation, so there exists some $x \in \{1..n\}$ that appears more than once in θ. Suppose that the last appearance of x is in position k. By Lemmas 3.11 and 3.12, replacing this appearance of x by y (where $y \in \{1..n\}$ does not appear in θ) yields a word δ that is hit by strictly fewer permutations in S_n. Since the cop performs a random permutation every n moves, each permutation is equally likely. Therefore $p_\delta > p_\theta$, which contradicts the maximality of p_θ. Therefore, $p_\omega > p_\theta$ for all $\omega \in S_n$ and $\theta \notin S_n$.

□

Note that if ω is a random permutation then p_ω is equal to the number of derangements of ω, which is $d_n = n! \sum_{k=0}^{n} \frac{(-1)^k}{k!}$, so the probability that a random permutation is a derangement of ω is $d_n/n! = \sum_{k=0}^{n} \frac{(-1)^k}{k!} \approx 1/e$.

Therefore, for a robber using permutation ω, we have that it takes the DFP cop $1/(1-p_\omega)$ iterations to capture him; therefore the expected capture time is bounded by $n/(1-p_\omega) = n/(1 - d_n/n!)$. However, the expected point of capture in the final iteration is not all the way at the end, as is discussed in the following lemmas.

Lemma 3.13. *Given that the robber is captured during a given iteration of permutation search, the capture occurs at*

$$\frac{(n+1)!}{n! - D_n} \sum_{j=2}^{n+1} \frac{(-1)^j (j-1)}{j!},$$

where D_n is the number of derangements in S_n.

Proof. Theorem 3.5 in [36] states that $\sum_{k=1}^{n} kF_{n,k} = (n+1)! \sum_{j=2}^{n+1} \frac{(-1)^j (j-1)}{j!}$, where $F_{n,k}$ is the number of permutations in S_n in which the first fixed point is at k. (Note that this is equivalent to the number of permutation strategies of the robber which give capture time k for a given permutation strategy of the cop.) The number we are looking for, then, is the expected first fixed point given that a fixed point exists. Therefore we divide by $n!-D_n$, the number of permutations in S_n containing a fixed point. □

Lemma 3.14. *Given that the robber is captured during a given iteration of permutation search, the asymptotic value of the expected capture time within this iteration* $\frac{e-2}{e-1}n$.

Proof. Theorem 3.6 in [36] gives the asymptotic value of the numerator of the sum in Lemma 3.13 to be $\frac{e-2}{e}(n+1)!$ (plus some lower order terms). Therefore, we have that the asymptotic value of the expected capture time in an iteration given that capture occurs during this iteration is

$$E = \frac{\frac{e-2}{e}(n+1)!}{(1-1/e)n!} = \frac{e-2}{e-1}n.$$

□

Therefore, we have the following bound on expected capture time.

Theorem 3.4. *The expected capture time in the cop vs robber game on K_n tends to n for large n.*

Proof. The expected capture time is $n/(1-d_n/n!)-(n-E)$, which tends to

$$\left(\frac{e}{e-1} - 1 + \frac{e-2}{e-1}\right) n = n$$

□

Note that we know that n is indeed the value of this game, since both players can guarantee expected capture time n by doing independent uniform movement at each step (as mentioned at the beginning of the section). Therefore we see that, asymptotically, the DFP strategy does quite well for the cop on K_n even against a less restricted adversary.

3.3 Depth First Pursuit on a Binary Tree

In this section we discuss the efficacy of the uniform depth first pursuit strategy on a (complete) binary tree of height h. Note that $n{=}2^{h+1}{-}1$ on this tree. Suppose that our cop begins her depth first pursuit at the root (the vertex at height 0). Though this strategy worked well in our previous examples, both against an immobile hider and on K_n against a mobile hider, we shall see that the robber can take advantage of the predictability of this strategy on the binary tree. The value of this game is n [23], but we shall see that the robber can guarantee an expected capture time of $\Omega(n^2)$ against a cop performing a uniform DFP strategy.

Consider the following robber strategy: the robber begins at a uniformly random leaf and stays there until the cop arrives at a height $h{-}1$ vertex for the first time; if he is not caught, he goes directly to the root and remains there until the second to last time at which the cop visits height 1 before returning to the root, at which point he stays two levels below the cop until the last time the cop visits $h-1$ before returning to the root; and so on, alternating between going to a leaf and staying there until it's time to go directly to the root and stay there, so he can only be caught twice per main branch, with a probability of $2^{-(h-1)}$ each time. We make this strategy precise and hopefully less confusing, and provide an example for small h (see Table 3.3) below.

- $t = 0$: Robber begins at a uniformly random leaf (height h).
- $1 \leq t \leq h{-}2$: Robber stays at his leaf.
- $h{-}1 \leq t \leq 2h{-}2$: Robber starts going directly to the root, so that he is at height $h{-}(i{+}2)$ at time $h{+}i$ for all $i \in \{-1, 0, 1, \ldots, h{-}2\}$.
- $2h{-}2 \leq t \leq 2^h$: Robber stays at root.
- $2^h \leq 2^{h+1}{-}(h{+}2)$: Robber starts following a uniform DFP but stays at two levels above the level prescribed by a uniform DFP.
- $t = 2^{h+1}{-}(h{+}1)$: Robber is at height $h{-}1$.
- $2^{h+1}{-}h \leq t \leq 2^{h+1}{-}2$: Robber is at a leaf.
- $2^{h+1}{-}2 \leq t \leq 2(2^{h+1}{-}2)$: Repeat the above steps.

Note that meanwhile, the DFP strategy prescribes that the cop be at the following heights at these times:

- $t = 0$: Cop begins at the room (height 0).
- $1 \leq t \leq h{-}2$: Cop is at height t at time t.
- $h{-}1 \leq t \leq 2h{-}2$: Cop is at height $h{-}1$ at time $h{-}1$ and at height h at time h. At times h through $2h{-}2$ she is therefore at a height strictly greater than the robber's.
- $2h{-}2 \leq t \leq 2^h$: Cop does not return to the root until time $2^{h+1}{-}2$, so she is still at a height strictly greater than the robber's.

- $2^h \leq 2^{h+1}-(h+2)$: Cop is at a height exactly two greater than the robber's, by the robber's design.

- $t = 2^{h+1}-(h+1)$: Cop is at height $h-1$.

- $2^{h+1}-h \leq t \leq 2^{h+1}-2$: Cop is at a height strictly smaller than that of the robber.

- $2^{h+1}-2 \leq t \leq 2(2^{h+1}-2)$: Repeat the above steps.

Therefore, there are only four times during each iteration of a DFP at which the players are at the same height.

As an example, Table 3.3 below lays out the cop's and robber's heights at time t on a height 4 tree, for all $t \leq 29$ (the number of steps it takes to complete a depth first pursuit on one main branch of a tree of height 4).

Table 3.1: A table of the cop's and robber's heights during the first half of a DFP on a tree of height 4 (where $p_t = \mathbb{P}(\text{capture at } t)$).

t	$c(t)$	$r(t)$	p_t	t	$c(t)$	$r(t)$	p_t
0	0	4		15	1	0	
1	1	4		16	2	0	
2	2	4		17	3	1	
3	3	3	2^{-3}	18	4	2	
4	4	2		19	3	1	
5	3	1		20	4	2	
6	4	0		21	3	1	
7	3	0		22	2	0	
8	2	0		23	3	1	
9	3	0		24	4	2	
10	4	0		25	3	1	
11	3	0		26	4	2	
12	4	0		27	3	3	2^{-3}
13	3	0		28	2	4	
14	2	0		29	1	4	

Lemma 3.15. *The expected capture time for a cop doing a uniform depth first pursuit on a binary tree against the robber doing the strategy defined above is greater than*

$$\frac{(n-15)(n-1)}{8}.$$

Proof. Using the strategy defined above, the robber is caught during one depth first pursuit with probability no greater than $4(2^{-(h-1)})$. Each depth first pursuit is independent of the previous ones, and each of the expected $\frac{2^{h-1}}{4} - 1$ unsuccessful depth first pursuits visits each edge twice, so takes $2(n-1) = 2(2^{h+1}-2)$ steps to complete. Therefore the cop expects to spend at least time

$$\left(\frac{2^{h-1}}{4} - 1\right)(2(2^{h+1}-2)) \quad = \quad \left(\frac{2^{h-1}-4}{2}\right)(2^{h+1}-2)$$

$$= \left(\frac{2^{h+1} - 16}{8}\right)(2^{h+1} - 2)$$
$$= \frac{(n - 15)(n - 1)}{8}$$

not capturing the robber. □

Since the actual value of the game is n [23], this is quite an unimpressive result for the cop. However, the cop can—by modifying her depth first pursuit strategy only slightly—make such treachery more difficult for the robber. For instance, randomizing her starting time may already render the robber strategy defined above less effective.

Chapter 4

Hunters & Moles

In this variation, which comes from Dick Hess [29] (who heard it from Simon Bexfield), the cop and the robber once again move simultaneously. The robber is constrained by the edges of the graph as usual. The cop is not and can move around freely—however, she cannot see her adversary. We characterize the class of graphs on which a single cop (whom we will call the "hunter") is guaranteed to catch the robber (let's call him a "mole") in bounded time. We will call such graphs hunter-win. Any graphs that are not hunter-win will be called mole-win.

4.1 A Characterization of Hunter-Win Graphs

Note first that since a winning strategy for the hunter must **guarantee** capture in bounded time (against any possible trajectory undertaken by the mole), it is equivalent to consider this game played by a hunter and an omniscient, adversarial mole—i.e. a mole who makes all the "worst-case scenario" moves (from the hunter's perspective) to maximally increase the length of the game. We will be considering this adversary in the remainder of this chapter.

4.1.1 Some Hunter-Win Graphs

We begin by showing that certain kinds of trees are hunter-win. We will see in Theorem 4.1 that these are the only hunter-win graphs, and give a forbidden subgraph characterization of them.

Lemma 4.1. *The path P_n is hunter-win for all n.*

Proof. Suppose the path has vertices $v_1, v_2, \dots, v_n$. There are two possibilities for the mole—(1) he is an **odd mole**, meaning that he is at odd-numbered vertices on odd moves (i.e. his initial position is on an odd vertex) or (2) he is an **even mole**, defined accordingly. Suppose first that we are playing against an even mole. Then at time 1, if the hunter chooses vertex v_2, the mole must be at a vertex in $\{v_4, v_6, \dots\}$. At time 2, the hunter chooses vertex v_3, limiting the mole's choices to the set $\{v_5, v_7, \dots\}$. At time 3, the hunter chooses v_4, and the mole is limited to $\{v_6, v_8, \dots\}$. Following this strategy, when the hunter goes to vertex v_j, the mole must be at a vertex in the set $\{v_{j+2}, v_{j+4}, \dots\}$, and so the hunter is continually limiting the mole's possible locations until finally she goes to v_{n-2} and forces the mole onto v_n. Then she heads to v_{n-1}, knowing that the mole will be forced to join her there on his next turn. Using this strategy against the even mole, the hunter ensures capture in $n-2$ turns.

Now suppose that she is playing against an odd mole. After the $n-2$ turns described above, she is at vertex v_{n-1}, and having not captured the mole at this point, it is clear that the mole had started at an odd vertex. Now the hunter can stay at v_{n-1} for a turn. The mole is constrained to the vertices $\{v_1, v_3, \dots, v_{n-3}\}$ if n is even and $\{v_2, v_4, \dots, v_{n-3}\}$ if n is odd, and the hunter proceeds using the same strategy as against the even mole and wins in an additional $n-2$ turns, for a total guaranteed capture time of at most $2n-4$. □

Definition 4.1. A **lobster** is a tree containing a path P such that all vertices are within distance 2 of P. A path satisfying this condition will be called a **central path**. A **knee** of a lobster with central path P is a non-leaf vertex at distance exactly 1 from P. For a knee a in a lobster G, define its associated **hip** to be the vertex v such that $v \in N(a) \cap P_m$. Any leaf adjacent to a knee is called a **foot**.

Lemma 4.2. *All lobsters are hunter-win.*

Proof. Let G be a lobster. As in Lemma 4.1, we will define a notion of an odd mole (even moles are defined analogously). Let $P = v_1, \ldots, v_m$ be a longest central path on G. We define an **odd mole** on G to be a mole who begins either

(*a*) on an odd vertex of P,

(*b*) on a knee adjacent to an even vertex of P,

or, (*c*) on a foot at distance 2 from an odd vertex of P

We will refer by V_j^1 to the set of knees adjacent to hip v_j, and similarly by V_k^2 to the set of feet adjacent to a knee in V_k^1.

We will suppose first that the hunter is playing against an odd mole and note that as in Lemma 4.1, if the mole is not caught at the end of the following scheme, then the hunter will know that the mole was in fact an even mole, and can repeat the scheme starting from the opposite end of P.

The hunter starts on v_1. Let M_t be the set of vertices to which the mole may move at time t in order to avoid being caught in one move. We proceed by induction on t

to show that when the hunter is on v_i for the first time (say at time t), M_t consists of the vertices in

$$\begin{aligned} &(a) \quad \{v_j : j \in \{i{+}2, i{+}4, \dots\}\}, \\ &(b) \quad V_k^1 \text{ for some } k \in \{i{+}1, i{+}3, \dots\}, \\ \textit{and} \quad &(c) \quad V_\ell^2 \text{ for some } \ell \in \{i, i{+}2, \dots\} \end{aligned}$$

When $t = 1$, since the hunter is on v_1, M_1 consists of

$$V_3 \cup V_5 \cup \cdots \cup V_{2k-1} \cup V_2^1 \cup V_4^1 \cup \cdots \cup V_{2k}^1 \cup V_3^2, V_5^2 \dots$$

where $2k{-}1 = n{-}1$ if n is even and $2k{-}1 = n$ if n is odd. (Note that $V_1^1 = V_1^2 = \emptyset$ by the maximality of P). Now suppose that for some t, the hunter is at v_i and M_t is as described in the claim above.

If $V_i^2 \neq \emptyset$, then the hunter moves to a vertex in V_i^1 at time $t{+}1$ to check that the mole is not on this branch. If the mole is in V_i^2, then the hunter can simply check every knee in V_i^1 by alternatingly being on a knee in V_i^1 and on vertex v_i, until all knees in V_i^1 have been checked. The mole cannot leave this branch while the hunter checks it in this way, since when she checks V_i^1, he also goes to a vertex in V_i^1, and therefore when she goes to v_i, he must go to a vertex in V_i^2. Note that this also means that the mole cannot switch knees, so once the hunter searches a vertex in V_i^1, there is no possibility of recontamination. Therefore the number of moves that the hunter spends searching through V_i^1 is equal to $2k_i$ where k_i is the number of knees adjacent to v_i. Meanwhile, if the mole was at v_{i+2} at time t, he can move to v_{i+1} or v_{i+3}; even while the hunter is busy checking her current branch, he cannot get back to v_i since the hunter is in V_i^1 when the mole is on v_{i+1}, and so her next move is to v_i, forcing the mole back. If the mole was in V_{i+1}^1, then he can move to v_{i+1} or to a vertex in V_{i+1}^2. Finally, if the mole was in V_{i+2}^2, he moves to a vertex in V_{i+2}^1. Then the hunter returns to v_i at time $t+2k_i$ and $M_{t+2k_i} = M_t \backslash V_i^2$. At time $t+2k_i+1$, the hunter moves to v_{i+1} and following the same arguments as above, we can see that M_{t+2k_i+1} consists of (a) $v_{i+3}, v_{i+5}, \ldots$, (b) $V_{i+2}^1, V_{i+4}^1, \ldots$, and (c) $V_{i+1}^2, V_{i+3}^2, \ldots$.

If instead $V_i^2 = \emptyset$ then the hunter moves to v_{i+1}. If at time t the mole were at v_{i+2}, he must now move to v_{i+3} or a vertex in V_{i+2}^1 to avoid immediate capture. If he was in V_{i+1}^1, he must move into V_{i+1}^2 (therefore if $V_{i+1}^2 = \emptyset$ then $V_{i+1}^1 \not\subset M_t$ since he would be captured at time $t+1$ on v_{i+1}). And finally, if he was in V_{i+2}^2, he must move into V_{i+2}^1. Therefore, M_{t+1} consists of (a) $v_{i+3}, v_{i+5}, \ldots$, (b) $V_{i+2}^1, V_{i+4}^1, \ldots,$, and (c) $V_{i+1}^2, V_{i+3}^2, \ldots$.

Note that according to this scheme, if $V_i^2 = \emptyset$, then (regardless of whether or not V_i^1 is empty) the hunter moves to v_{i+1} from v_i. Therefore, if the hunter gets to vertex v_{m-1} using this scheme without having captured the mole, she knows that the odd mole must be at a vertex in V_{m-1}^2; the hunter wins by checking the vertices in V_{m-1}^1 as described above. Otherwise, the hunter's adversary was an even mole, and the scheme repeats identically in the opposite direction, ending in capture at the other end of P. □

4.1.2 Mole-win Graphs

Lemma 4.3. *Let H be any mole-win graph. Then any graph G containing H as a subgraph is also mole-win.*

Proof. Let H be a mole-win graph and G a graph containing H as a subgraph. Then the mole has a strategy, $\mathcal{R}$, on H such that no strategy of the hunter can guarantee capture in bounded time. When the game is played on G, the mole still performs strategy $\mathcal{R}$. The hunter can now use the vertices of $G \backslash H$, but since the mole's strategy has him staying on the vertices of H, she will not find the mole there. Since she was already free, in the game played on H, to go to any vertex at any time (and also to wait there as long as she likes), using the vertices of $G \backslash H$ does not offer any advantage in beating strategy $\mathcal{R}$. Therefore, G is mole-win. □

We now show that we can focus our attention on trees.

Lemma 4.4. *The cycle C_n is mole-win for all $n \geq 3$.*

Proof. Suppose the cycle has vertices $v_1, v_2, \ldots, v_n$. Note first that the hunter does not benefit from knowing the history of the mole's moves. If the two players occupy two different vertices (say, the hunter is on v_i and the mole on v_j), then the mole has two choices of moves—either to v_{j-1} or to v_{j+1}. Therefore the hunter has a positive probability of making the wrong choice (i.e. the prescient mole will be able to slip away by choosing a neighboring vertex different from the choice of the hunter). Therefore, if the two players occupy two different vertices at turn t, the mole can ensure that they will occupy two different vertices at turn $t+1$. □

Figure 4.1 shows a visualization of this argument. In the case of the cycle, it is rather trivial, but will be useful in helping to visualize the argument in Lemma 4.5. We label the vertices with "+" if that vertex is a possible choice for the mole, and "0" otherwise. The arcs represent move sequences that the hunter can choose. In the case of the cycle, this diagram shows that no matter what choice the hunter makes for her move at time t, the situation at time $t+1$ will be the same as at time t.

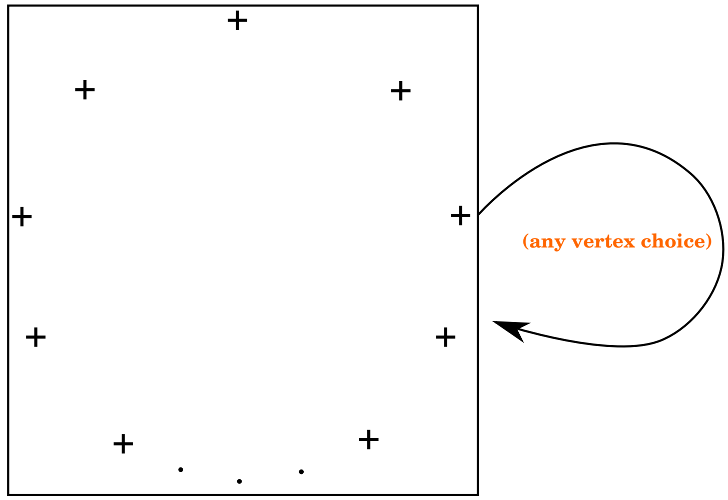

Figure 4.1: A diagram of the mole's choices on C_n given any hunter move sequence

Lemmas 4.3 and 4.4 immediately yield the following corollary.

Corollary 4.1. *Any graph containing a cycle is mole-win.*

We now show that by far not all trees will be hunter-win, either. In the remainder of this chapter, let S be the spider graph with three legs of length three. For notational convenience, we will denote the hub vertex by 0, and the vertices on the legs by v_1, v_2, v_3 where v is one of the symbols a, b, c, and for all v, $0 \in N(v_1)$, $v_1 \in N(v_2)$, and $v_2 \in N(v_3)$. The graph S is represented in Figure 4.2.

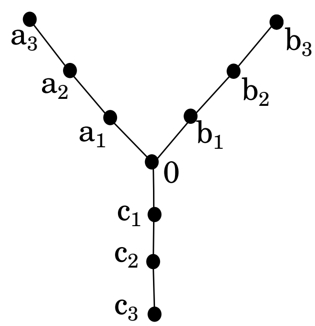

Figure 4.2: The graph S

Lemma 4.5. *S is mole-win.*

Proof. Suppose that the mole is an even mole (so that at time 1, he can be at vertices $0, a_2, b_2$, or c_2). We will show that the hunter cannot make progress against an even mole. To do this, we will go through every possible situation that may arise, and show that in every move sequence of the hunter, a previously occuring situation will repeat, showing that the mole never runs out of possible locations. (Note that the previously defined adversarial mole is caught at time t if and only if at time t, there are no possible places for a random mole to be such that he will not be guaranteed to be caught immediately by the hunter.)

We start with the mole at one of the following vertices: $0, a_2, b_2, c_2$. The hunter has (up to symmetry) two distinct choices that will further limit the mole's possible locations: to move to 0 or to move to (without loss of generality) a_2.

Sequence 1. The hunter moves to 0. Then if he is not to be caught immediately, the mole may start on a_2, b_2, or c_2. Therefore the mole may, on his next turn, be at a_1, a_3, b_1, b_3, c_1, or c_3. The hunter again has two choices (up to symmetry): she moves to a_1 or to a_3. In either case, the mole will be at one of the vertices $0, a_2, b_2, c_2$ on the next turn and we have repeated the initial scenario, showing that the hunter cannot win by initially choosing to be at vertex 0.

Sequence 2. The hunter moves to a_2. Then the mole's initial position is at $0, b_2$, or c_2. Therefore the mole will be at a_1, b_1, b_3, c_1, or c_3 next. The hunter now has three choices:

(a) The hunter goes to a_1.

(b) The hunter goes to b_1.

(c) The hunter goes to b_3.

Both options (b) and (c) result in the possible positions a_1, b_1, c_a, c_3 for the mole, and therefore he will next be at $0, a_2, b_2$, or c_2, resulting in the initial situation again.

If the hunter chooses option (a), then the mole must be at vertices b_1, b_3, c_1, or c_3, and therefore will next appear at vertices $0, b_2$, or c_2. The hunter's move sequence may take one of two turns now:

(i) The hunter goes to 0.

(ii) The hunter goes to b_2.

In move sequence (i), the mole must have been at b_2 or c_2 and therefore will next be at b_1, b_3, c_1, or c_3. The hunter may then move to either b_1 or b_3, but in either case, the mole will next be at $0, b_2$,or c_2, creating the same situation (where the hunter must choose between (i) and (ii)), so no progress has been made. In move sequence (ii), the mole may be at a_1, b_1, c_1, or c_3. The hunter can then go to a_1, so that the mole's next position is at $0, b_2$, or c_2 and again the hunter must choose between (i) or (ii), or the hunter can go to b_1 or b_3, both resulting in the mole appearing at one of $0, a_2, b_2, c_2$, and the hunter chooses between sequence 1 or 2.

So we see that for every choice that the hunter can make in determining her move sequence, she will eventually repeat a previously occuring situation, showing that she cannot get the mole into a position where he has no safe choices. □

As in the case of Lemma 4.4, we can represent this with the following figure.

In Figure 4.3, the top left leg is the leg containing the a_i vertices, and the bottom leg is the leg containing the c_i vertices. In this figure we include all of the possible hunter moves from any given position (including the ones that we did not discuss in

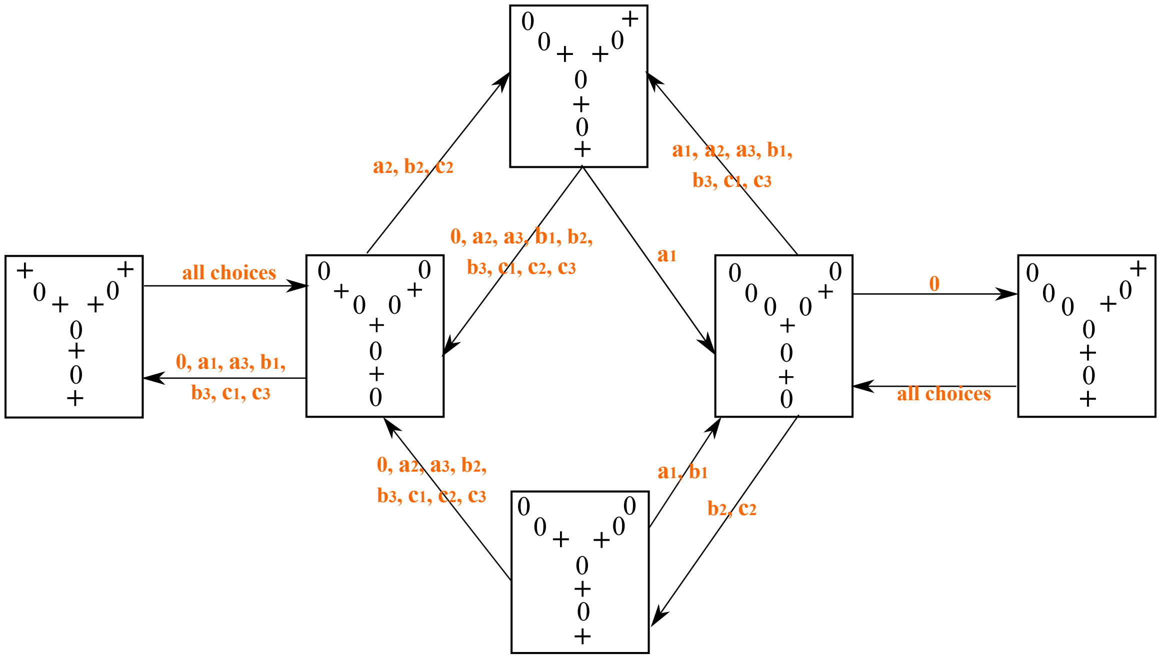

Figure 4.3: A diagram of the mole's choices given any hunter move sequence

Lemma 4.5 because they represent the hunter checking a vertex on which she knows the mole will not be). With this visualization, it is easy to see that there are only six distinct positions in which the hunter can find herself, and none of them include a guaranteed capture of the mole. Lemma 4.3 yields the following restriction on potential hunter-win trees.

Corollary 4.2. *Any tree containing S as a subgraph is mole-win.*

4.1.3 Characterization

We are now ready to give a forbidden subgraph characterization of lobsters, and to characterize all hunter-win graphs.

Lemma 4.6. *A tree G is a lobster (as defined in Definition 4.1) if and only if it does not contain S as a subgraph.*

Proof. Let G be a lobster containing a path P such that P is a longest path in G satisfying the condition that all of its vertices are within distance 2 of P. Suppose, for contradiction, that G contains S as a subgraph and let $v \in V(G)$. If $v \in P$ then v can be adjacent to at most two vertices that are also on P. Since all vertices are within distance 2 of P, v is not on a length three path disjoint from P, and so v cannot be the hub vertex of S. If instead $v \notin P$ then either v is a leaf (and therefore not the hub of S) or v is at distance exactly 1 from P, and then all but at most one of its neighbors are leaves (and again, v could not be the hub of S). Hence, no vertex of G is the hub of S, which is a contradiction.

Suppose now that G is any tree not containing S as a subgraph. Let P be a path of maximum length in G and label its end points x and y. Let v be any vertex in G that is not on P and let w be the vertex on P that is on the path connecting v to P. If $d(x, w) \leq 2$ then $d(v, w) \leq 2$ since otherwise the v—y path is longer than P. (Similarly, we have $d(v, w) \leq 2$ if $d(y, w) \leq 2$.) If both $d(w, x) > 2$ and $d(w, y) > 2$, then if $d(v, w) > 2$ then S would be a subgraph of G (with w as its hub). Therefore $d(v, w) \leq 2$ and so all vertices are within distance 2 of P, which makes G a lobster. □

Lemma 4.6, combined with Lemma 4.5, immediately yields the following corollary.

Corollary 4.3. *If G is hunter-win, then G must be a lobster.*

Finally, using Lemma 4.2 and Corollary 4.3, we have our desired theorem.

Theorem 4.1. *A graph is hunter-win if and only if it is a lobster.*

4.2 Optimality of the Hunter Strategy

In Lemmas 4.1 and 4.2, we described strategies for the hunter which checked for (without loss of generality) an even mole first, and then for an odd mole. On a path P_n on n vertices, the strategy took time $2(n-2)$ and on a lobster with a longest central path P_m on m vertices, containing k knees, the strategy took time $2(m-2)+4k$. It may seem, a priori, that considering these two types of moles separately may not be the most intelligent strategy for the hunter—perhaps she can do something that would be quicker. However, we shall see that the hunter does not, in fact, have a faster strategy for capturing the mole.

Lemma 4.7. *Let G be a lobster with k knees and with a longest central path P_m with m vertices. Any winning hunter strategy must visit each knee and each of the $m-2$ internal vertices of P_m at least twice.*

Proof. Suppose that there is a knee or internal vertex v_j that is never visited by the hunter on an odd turn. v_j has at least two neighbors. We will call one of the neighbors v_{j-1} and another v_{j+1}. Then consider the strategy ρ for the mole defined as follows

$$\rho_s = \begin{cases} v_j & \text{if } s \text{ is odd} \\ v_{j-1} & \text{if } s \text{ is even and } \sigma_s \neq v_{j-1} \\ v_{j+1} & \text{if } s \text{ is even and } \sigma_s = v_{j-1} \end{cases}$$

We claim that ρ beats σ (that is, the mole remains not caught at time M).

If σ were to beat ρ, then the hunter must catch the mole either (1) on v_j or (2) on $v_{j\pm1}$. Since σ never has the hunter visiting v_j on an odd turn, and ρ never has the mole visiting v_j on an even turn, case (1) will never occur. On even turns, ρ tells the mole to go to v_{j-1} if and only if the hunter is not there, and so he cannot be caught there. Similarly, the mole goes to v_{j+1} when and only when the hunter goes to v_{j-1}, and so he also cannot be caught there. (Note that since the hunter cannot occupy both v_{j-1} and v_{j+1} at the same time, the mole always has one of these two options at all even turns.)

Therefore, the hunter using strategy σ cannot beat the mole using strategy ρ on G. Since σ was chosen to be any hunter strategy in which there is an internal vertex appearing at most once, any winning hunter strategy must visit each internal vertex at least twice. $\square$

An analogous argument to that in the proof of Lemma 4.7 in the case of the general lobster is presented below.

Lemma 4.8. *Let v be a hip in a lobster G with k adjacent knees. Any winning hunter strategy $\sigma = \{\sigma_1, \sigma_2, \ldots, \sigma_M\}$ must visit v at least $2k+2$ times.*

Proof. Let the hip v have k knees, $a_1, a_2, \ldots, a_k$. By u and w we will refer to the neighbors of v that are on P_m. Now suppose that the hunter only visits v a total of $2k+1$ times. Then without loss of generality, v is only visited k times on odd turns. Suppose that the first odd visit to v occurs at time $2r+1$. Then for all $s < 2r+1$ we define

$$\rho_s = \begin{cases} v & \text{if } s \text{ is odd} \\ u & \text{if } s \text{ is even and } \sigma_s \neq u \\ w & \text{if } s \text{ is even and } \sigma_s = u \end{cases}$$

Now suppose that we have defined ρ_s for all $s < 2m+1$ for some m where $\sigma_{2m-1} \neq v$ and for some j, $\sigma_{2m+1} = \sigma_{2m+3} = \cdots = \sigma_{2m+2j-1} = v$ (i.e. a string of j sequential odd-turn visits to v begins at time $2m+1$). Note that there exists some

$N(v)\backslash\{\sigma_{2m+1}, \sigma_{2m+3}, \ldots, \sigma_{2m+2j-1}\} \neq \emptyset$ since $|N(v)| \geq k+2$ and $j \leq k$. Let x be a vertex in this set, and let $y \neq x$ be neighbor of x.

Then for all $s \in [2m+1, 2m+2j+1]$ we define

$$\rho_s = \begin{cases} x & \text{if } s \text{ is even} \\ y & \text{if } s \text{ is odd and } s \neq 2m+2j+1 \\ v & \text{if } s = 2m+2j+1 \end{cases}$$

This yields a mole strategy $\rho = \{\rho_1, \rho_2, \ldots, \rho_M\}$ which beats hunter strategy σ since by the definition of ρ, we have $\rho_s \neq \sigma_s$ for all $s \in [M]$. □

Lemmas 4.7 and 4.8 combine to tell us that every knee must be visited twice, any internal path vertex that isn't a hip must be visited twice, and any hip requires an additional number of visits equal to twice the number of knees adjacent to it. Therefore, we have as an immediate corollary that all winning hunter strategies on a lobster G containing k knees and a longest central path on m vertices must take time at least $2(m-2)+4k$.

Theorem 4.2. *The hunter strategy defined in Lemma 4.2 is an optimal strategy.*

4.3 Hunter vs. Sneakier Mole

In our proof of Lemma 4.2, we relied heavily on the fact that the mole was forced to move at each step by making use of the fact that the mole has a fixed parity. What if this were no longer the case? We represent this variation by keeping the rules of the game entirely the same, but adding loops to the graph at vertices at which the mole is allowed to sit.

This addition is quite unfortunate for the hunter, as can be seen in Lemma 4.9, below.

Lemma 4.9. *Any graph containing at least two loops is mole-win.*

Proof. First fix $n \geq 2$ and consider the graph P_n with a loop at both of its endpoints, with vertices labeled $1, 2, \ldots, n$ from left to right. Its diagram will behave very similarly to the diagram in Figure 4.1—any choice of first move for the hunter will return the mole to his initial situation:

> Regardless of the hunter's choice of initial position, all of the vertices will be adjacent to a vertex not occupied by the hunter, and therefore fair game for the mole. In particular, if the hunter chooses any vertex v with a label from $\{3, 4, \ldots, n-2\}$, then all vertices are adjacent to either one or two vertices not occupied by the hunter (i.e. to a vertex represented by a "+" in the diagram). If the hunter chooses vertex 2 or $n-1$, then any internal vertex is adjacent to a "+" and since the hunter is not occupying the endpoints and they are adjacent to themselves, they also have a "+" in the diagram. Finally, if the hunter begins at an endpoint, all vertices in $\{3, 4, \ldots, n-2\}$ are adjacent to two "+" vertices, and vertices $1, 2, n-1$, and n are adjacent to one "+" vertex.

By Lemma 4.3, any graph containing as a subgraph P_k with a loop at each end point (for any $2 \leq k \leq n$) is mole-win. □

So then the question of a graph with loops becomes the question of a graph with a single loop. We will see that the mole can make good use of a single loop, but only in certain placements.

Lemma 4.10. *Any path P_n with a loop at a vertex that is at distance less than or equal to 2 from an endpoint is hunter-win (for all $n \geq 1$).*

Proof. Label the vertices of P_n from 1 to n moving left to right, and first place the loop at the vertex labeled "1." We claim that the following sequence of moves for the hunter will guarantee capture of the mole in $2(n-1)$ steps:

$$\mathcal{H} = \{n-1, n-2, \ldots, 2, 1, 1, 2, 3, \ldots, n-1\}.$$

In her movement from right to left, the hunter is progressively taking away options. For his first move, the mole can be anywhere but at n; for his second move, anywhere but at $n-1$; for his third move, anywhere but n and $n-2$; for his fourth, anywhere but $n-1$ and $n-3$; and so on. By the time she gets to 1, the mole's options are therefore only the odd vertices (that is, the diagram looks like $\boxed{+0+0+\cdots}$. By waiting one more turn at 1, the hunter turns the mole's diagram into $\boxed{0+0+0\cdots}$ and so as she moves back along the path to the right, she is taking away (at every other step) one more position of the appropriate parity, ensuring that the situation is just like in the original game on P_n with no loop.

Now suppose that the loop is at vertex 2 and consider the hunter strategy

$$\mathcal{H} = \{n-1, n-2, \ldots, 3, 2, 2, 3, \ldots, n-1\}.$$

An identical argument to the one above shows that by the time the hunter reaches 2 for the first time, the mole's diagram looks like $\boxed{0+0+0\cdots}$, and so by remaining at 2 for one more step and then continuing to the right, the hunter progressively removes one more position of the correct parity at every other step.

Finally, if the loop is at vertex 3, the hunter strategy

$$\mathcal{H} = \{n-1, n-2, \ldots, 3, 2, 3, 3, 2, 3, \ldots, n-2, n-1\}$$

behaves nearly the same way as in the previous arguments. When the cop gets to vertex 3 for the first time, the mole's diagram looks like $\boxed{+ + +0+0+\cdots}$, so that following the move sequence to $3, 2, 3, 3$, the diagram looks as in Figure 4.4.

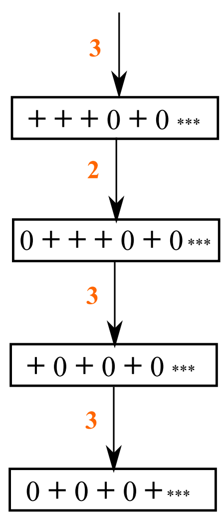

Figure 4.4: A diagram of the mole's choices after the hunter's first arrival at the loop

Now as in the previous two arguments, the hunter is progressively removing one more position of the correct parity from the mole's list of options at every other step. □

Lemma 4.11. *Any lobster G with a maximal length central path P with a loop off of a vertex v (that is, the loop is either at v, at a knee adjacent to v, or at a foot at distance 2 from v) that is at distance less than or equal to 2 from an endpoint of P is hunter-win.*

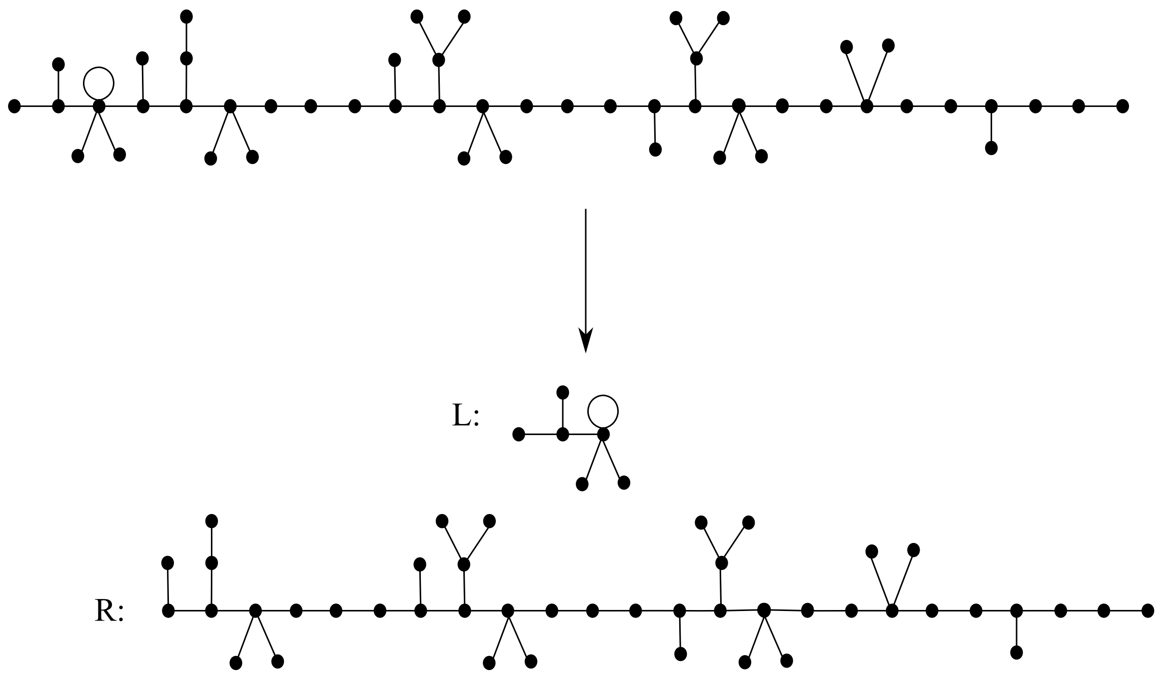

Figure 4.5: An example of a lobster with a loop, separated into L and R

Proof. Create two subgraphs, L and R of G, such that L is induced by v and all vertices on the path between it and the nearest path endpoint (and all neighbors of these vertices), and R is $G \backslash L$, as in the example of Figure 4.5. Since G without the loop is hunter win by Theorem 4.1, R is hunter-win by the contrapositive of Lemma 4.3. Therefore, following the first half of the strategy described in the proof of Lemma 4.2, the hunter can clear R for a mole of the proper parity (without loss of generality, suppose she clears it for an even mole first, and we call this sequence of moves R_E, and let R_O be the reverse of R_E). Note that by using the loop, the mole can change his parity, but he cannot get into R as an even mole after R has already been checked, so when the hunter gets to the loop for the first time, the mole is either in L or is an odd mole in R.

There are precious few options for how L may look, up to symmetry:

(1) If v is an endpoint of the path, then $L=\{v\}$, and so the hunter wins with strategy $R_E - v - v - R_O$, much like in the first case of Lemma 4.10.

(2) If v is a vertex of the central path at distance 1 from the endpoint, then L is a star with v at its hub. Then once again the strategy $R_E - v - v - R_O$ captures the mole. Notice that if v is on a vertex off of the central path, adjacent to a vertex at distance 1 from the endpoint, then we could redefine the path so that v is on it.

(3) If v is on the path at distance 2 from the nearest endpoint, then either v has an off-path neighbor at distance 2 or it does not.

(i) v does not have an off-path neighbor at distance 2.

Label the on-path neighbor of v in L by w and the endpoint of the path by x. Then when the hunter goes through R_E and hits v for the first time, the mole could go to x, w, an off-path neighbor of w, or v. Then the hunter goes next to w, so that the mole's options in L become w or an off-path neighbor of v. Now the hunter goes back to v so that the mole's options are x, an off-path neighbor of w, or v, and so when the hunter stays at v for one more step, the mole's options become only w or a vertex in R (of the not-yet-checked parity). Now the hunter simply turns around and heads back into R, removing one vertex of the appropriate parity at each move, just as in Lemma 4.10, and guaranteeing that the mole must be in R, and will be found at the end of the sequence $R_E - v - w - vv - w - v - R_O$.

(ii) The vertex v has an off-path neighbor at distance 2.

Label the k off-path neighbors which are **not** leaves (i.e. they are the middle vertex in a length 2 path starting at v and ending off of the central path) of v by $a_1, a_2, \ldots, a_k$. Notice that by maximality of the central path, all of the off-path neighbors of w are leaves. Now once the hunter completes R_E and hits v for the first time, the mole can go to any vertex in L for his next move. So the hunter can continue to cut off one more vertex of the proper parity at each turn, as before, by doing the following strategy:

$$R_E - v - w - v - a_1 - v - a_2 - v - \cdots - v - a_k - v - v$$

At this point, the mole must be at $a_1, a_2, \ldots$ or a_k. Now by going to $a_1 - v - a_2 - v - \cdots - v - a_k - v - w, v$, the hunter guarantees that the mole is in R, and so captures him during the rest of her sequence: R_O.

If v a knee adjacent to a hip at distance 2 from the nearest path endpoint, then this situation can be made identical to the one in (2), and if v is on a foot at distance 2 from a hip which is itself at distance 2 from the nearest path endpoint, then a redefinition of the path yields the situation in (1).

Therefore, we have considered every possibility for L, and shown that they are all hunter-win. □

In all of the arguments in the proof of Lemma 4.10, we again make use of the fixed parity of the mole. Even though he can stay at the loop for a while, he cannot change his parity in an effective way (and get "behind" the cop on vertices of the same parity she already checked). Far from all graphs (or even paths) with a single loop are hunter-win, however—and it stands to reason that the graphs which fail to be hunter-win are precisely those on which the hunter cannot clear the graph to one side of the loop while simultaneously keeping the mole from switching his parity in an effective way.

Lemma 4.12. *Any graph containing as a subgraph a copy of one of the following graphs is mole-win.*

G_1: *P_7 with a loop at its middle vertex.*

G_2: *P_7 with one looped non-path vertex, attached to the middle vertex.*

G_3: *P_7 with a path of length 2 attached to the middle vertex, with a loop at its endpoint.*

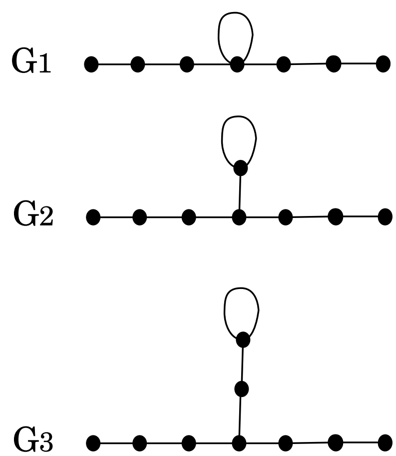

Figure 4.6: The graphs G_1, G_2, and G_3

Proof. G_1: Label the vertices of G_1 from (without loss of generality) left to right with $1, 2, \ldots, 7$ (so that the loop is at 4). By performing an analysis as in the proof of Lemma 4.5, one can see that the diagram of possibilities for the mole is exactly the diagram in Figure 4.7 (with symmetric situations identified on the diagram).

G_2: Label the vertices of G_2 from left to right on the path with $1, 2, \ldots, 7$ again, and label the remaining neighbor of 4 with an 8. The diagram of position possibility states for the mole is the one in Figure 4.8

G_3: Finally, label the vertices of G_3 from left to right on the path with $1, 2, \ldots, 7$ again, and label the remaining neighbor of 4 with an 8, and the remaining neighbor of 8 with a 9. The diagram of position possibility states for the mole is the one in Figure 4.9.

In all three cases, no move sequence of the hunter can guarantee capture of the mole. And consequently by Lemma 4.3, no graph containing G_1, G_2, or G_3 as a subgraph can be hunter-win.

□

The previous two lemmas immediately yield the following characterization of hunter-win graphs (in which we allow loops).

Theorem 4.3. *A graph is hunter-win if and only if it satisfies all of the following:*

(a) It is a lobster.

(b) It contains no more than one loop.

(c) If it does contain a loop, the loop is either

(i) at a vertex on the central path which is within distance 2 of the nearest endpoint, or

(ii) at a vertex off the central path which is a neighbor of such a path vertex.

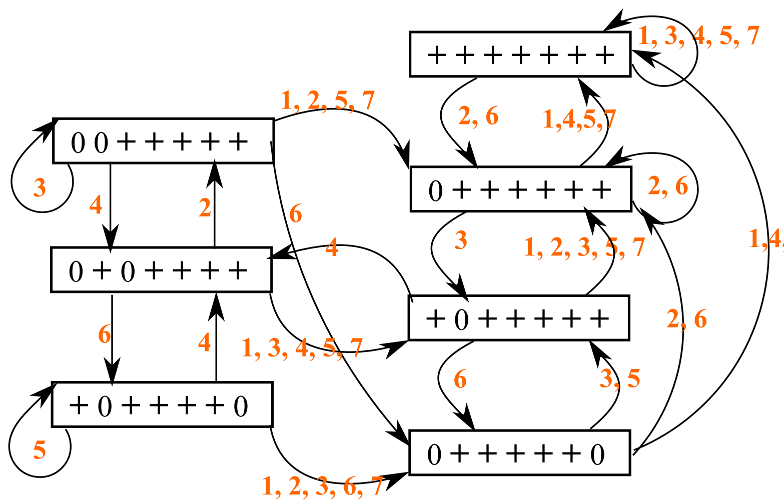

Figure 4.7: A diagram of the mole's choices in G_1 given any hunter move sequence

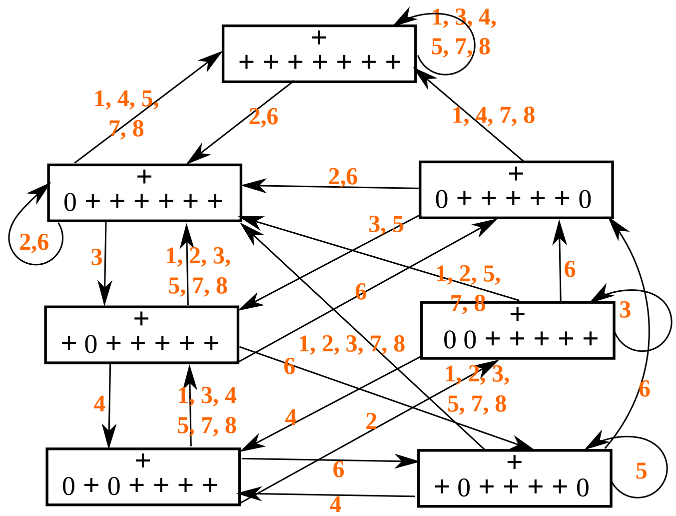

Figure 4.8: A diagram of the mole's choices in G_2 given any hunter move sequence

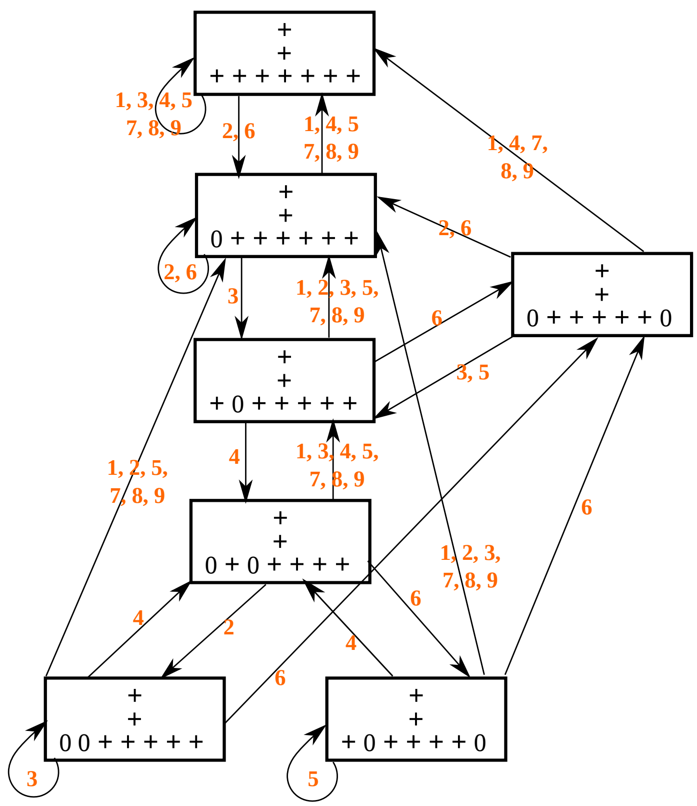

Figure 4.9: A diagram of the mole's choices in G_3 given any hunter move sequence

4.4 Appendix to Chapter 4

In this appendix we give an algebraic interpretation of the game studied in Chapter 4 and give some Matlab code that generates all winning hunter strategies for a graph given its adjacency matrix. The reader should be warned that the number of winning hunter strategies may be quite large for some graphs!

4.4.1 Hunter vs. Mole: An Algebraic Approach and Hunter-Win Strategy Generation

Let T be the adjacency matrix for the graph G. Let W_t be an n-dimensional vector corresponding to charges placed on the vertices, defined in the following way. Let $W_0 = \langle 1/n, 1/n, \ldots, 1/n \rangle$ and define $W_{t+1} = I_{\sigma(t)} T W_t$ where I_j is the $n \times n$ diagonal matrix with the (j, j) entry equal to 0 and 1's elsewhere on the diagonal.

Lemma 4.13. *A graph G is hunter-win if and only if there is a hunter strategy σ such that for some bounded M, $W_M = \mathbf{0}$.*

Example The Cycle. As one example of how we can use this algebraic interpretation, we can reprove Lemma 4.4 in the following way.

Lemma 4.14. *The cycle C_n is mole-win.*

Proof. Consider the cycle on n vertices, C_n. The random walk on C_n has transition matrix

$$\begin{pmatrix} 0 & 1/2 & 0 & \cdots & 0 & 1/2 \\ 1/2 & 0 & 1/2 & 0 & \cdots & 0 \\ 0 & 1/2 & 0 & 1/2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 1/2 & 0 & 1/2 \\ 1/2 & 0 & \cdots & & 1/2 & 0 \end{pmatrix}$$

Suppose that for some t, $W_t = \langle p_1, p_2, \ldots, p_n \rangle$, such that at least two of the entries in W_t are nonzero. We will show that there is no k such that $I_k T W_t = \mathbf{0}$.

$$TW_t = \langle \frac{p_2 + p_n}{2}, \frac{p_3 + p_1}{2}, \frac{p_4 + p_2}{2}, \ldots, \frac{p_n + p_{n-2}}{2}, \frac{p_1 + p_{n-1}}{2} \rangle$$

Suppose that for some k, $I_k T W_t = \mathbf{0}$. Note that for each i, p_i appears in two entries in W_t: one in which it appears with p_{i+n-2n} and another in which it appears with $p_{i-(n-2)}$ (with subscripts taken modulo n). Note that $i - (n-2) \equiv i + n - 2 \mod n$ if and only if $n = 4$. For now we suppose that $n \neq 4$. Then if, say, $p_i, p_j \neq 0$ (where $i \neq j$), TW_t will be a vector containing two instances of $p_i/2$ and two instances of $p_j/2$, and consequently setting any single entry of TW_t to zero will result in a non-zero vector. So therefore if at least two entries of W_t are non-zero, then $I_k T W_t \neq \mathbf{0}$ for all $k \in \{1..n\}$. Note also that if at least two entries of W_t are non-zero then $T I_k T W_t$ will have at least two non-zero entries for any choice of k.

For the case where $n = 4$, note that T is the matrix

$$\begin{pmatrix} 0 & 1/2 & 0 & 1/2 \\ 1/2 & 0 & 1/2 & 0 \\ 0 & 1/2 & 0 & 1/2 \\ 1/2 & 0 & 1/2 & 0 \end{pmatrix}$$

and so for any t, if W_t has no non-zero entries then $I_{\sigma(t)}TW_t$ has exactly one non-zero entry and if W_t has exactly one non-zero entry then $I_{\sigma(t)}TW_t$ has exactly one non-zero entry, yielding no way to get to $W_M = \mathbf{0}$ for any M. □

With this algebraic interpretation in mind, the following Matlab code defines a recursive function used to print all of the hunter-win strategies, if any exist (and will print nothing if the graph corresponding to T is mole-win). This function takes in an adjacency matrix T, a vector v (initially all 1s), a list VS of visited states (initially just a single entry—the vector of all 1's, and an ordered list of the vertices chosen for a winning hunter strategy (initially the empty list). It prints "hunter win" and the winning strategy every time it hits upon one; it will print all of the winning strategies that don't repeat an already visited state. (Warning: the list of strategies may be quite large!) To run this code for a graph of size n with adjacency matrix T, run the following command in Matlab: `huntermole(T,ones(1,n),ones(1,n),[])`

4.4.2 Hunter vs. Mole: Winning Strategy Generating Code

```
function huntermole(T,v,VS,winning_strat)
[r,c] = size(VS); %r=num. of distinct visited states
```

```
%%%%this part changes v and VS%%%%
for i = 1:length(v)
    temp = v;
    temp(i) = 0; % hunter picks position
    temp = temp*T; % flow charges corresponding to mole options

    %just want to save if they are 0 or 1 so change all #s > 1 to 1
    for m = 1:length(v)
        if temp(m) > 1
            temp(m) = 1;
        end
    end

    % check if the hunter has won yet
    if all(temp == zeros(1, length(temp))) == 1
        %then this is the zero vector, so we are done
        disp('hunter win');
        [winning_strat;i] %displays the winning strategy
    else
        % not done yet
        found = 0; % "found" is going to be 1 if v was in VS, 0 otherwise

        for k=1:r
            found = found + all(temp==VS(k,:));
```

```
            %adds 1 to "found" score if v was already in VS
        end

        if found == 0
            % then v wasn't found yet so add it to VS and apply function to
            % this new vector
            huntermole(T, temp, [VS;temp],[winning_strat;i]);
        end
    end
end
end
```

Chapter 5

Cops & Gamblers

Consider the following version of the game of cops and robbers: the players play on a graph G with vertex set $V = \{v_1, v_2, \ldots, v_n\}$. The cop moves as in the original version of this game, but her adversary is now a "gambler" who is not constrained by the edge set of the graph. Instead, at the beginning of the game, he picks a probability distribution, $p_1, p_2, \ldots, p_n$, where $p_i = \mathbb{P}(R_t = v_i)$ (where R_t is the gambler's position at time t, so that the R_t are i.i.d. random variables). This distribution is known by the cop, and the gambler maintains the same distribution throughout the game, which as usual terminates when the two players occupy the same vertex. We seek to answer the question of how long, in expected time, this game will go on; and indeed, we answer the question with "no more than time n" on any graph—a rather surprising theorem!

5.1 Cop vs. gambler on P_n

Consider this game played on $G = P_n$, the path on n vertices, with vertices labeled v_1 through v_n from left to right, for the sake of consistent visualization. Let C_t be the cop's position at time t. Suppose that $C_0 = v_1$, and R_t (with $t \in \mathbb{Z}_{\geq 0}$) is chosen according to the distribution $p_1, \ldots, p_n$ on the vertices, as explained above.

Lemma 5.1. *The cop can capture the gambler in expected time less than or equal to n on P_n.*

Proof. In our strategy, the cop will be moving from v_1 toward v_n, stopping when p_i is large enough to allow it. Let us make this notion precise in the following manner.

Suppose that the cop is at v_i. Let $m_i = n - i + 1$ be the total number of vertices "ahead" of the cop (including the one she is currently occupying) and let c_i=p_i+ ... +p_n be the sum of the probabilities of the vertices that are "ahead" of the cop (including the one she is currently occupying). If the cop is at v_i, let T_i be the expected time from **now** until capture. We claim that $T_i \leq m_i/c_i$ for all $i \in \{1..n\}$ and proceed by induction on m_i. Note that when $i = 1$, this statement is equivalent to the statement in the lemma.

When $m_i = 1$, $i = n$, so the cop is at v_n. Here $c_n = p_n$, and therefore (since the waiting time is a geometric random variable with parameter p_n), we have $T_n = 1/p_n = m_n/c_n$. Now suppose that the statement holds for all m_j for $j > i$, and suppose that the cop has arrived at v_i. If $p_i \geq c_i/m_i$, we are done, since $T_i = 1/p_i \leq m_i/c_i$. Now suppose that $p_i < c_i/m_i$. With probability p_i, the cop captures the gambler at v_i (which takes time 1), and with probability $1-p_i$, she moves on to vertex v_{i+1}. Now $m_{i+1} = m_i - 1 < m_i$ so the expected capture time from this new position is bounded by $\frac{m_{i+1}}{c_{i+1}} = \frac{m_i - 1}{c_i - p_i}$. This yields the following inequality:

$$T_i \leq 1 + (1 - p_i)\left(\frac{m_i - 1}{c_i - p_i}\right)$$

We have (for $i > 1$):

$$\begin{aligned} p_i &< \frac{c_i}{m_i} \\ \Longrightarrow c_i(c_i - 1) &< m_i p_i (c_i - 1) \\ \Longrightarrow c_i(c_i - m_i p_i + m_i - 1 + p_i - p_i) &< m_i(c_i - p_i) \\ \Longrightarrow \frac{c_i}{c_i - p_i}(c_i - p_i + m_i(1 - p_i) - (1 - p_i)) &< m_i \\ \Longrightarrow c_i + \frac{c_i(1 - p_i)}{c_i - p_i}(m_i - 1) &< m_i \\ \Longrightarrow 1 + \frac{1 - p_i}{c_i - p_i}(m_i - 1) &< \frac{m_i}{c_i} \end{aligned}$$

Therefore $T_i < \frac{m_i}{c_i}$ whenever $p_i < \frac{c_i}{m_i}$, as desired. □

5.2 Cop vs. gambler on a tree

We now consider this game on any tree. Again, label the vertices $v_1, \ldots, v_n$ with index $i < j$ if the cop will reach v_i before v_j.

Lemma 5.2. *The cop can capture the gambler in expected time less than or equal to n on any size n tree.*

Proof. The cop proceeds with a strategy similar to that in the case of the path, but when she must make a choice of a branch of the tree, she chooses the branch with the highest average probability among its vertices. In particular, if she is at vertex

v_i and chooses the branch containing vertex v_j, then $\frac{c_j}{m_j} \geq \frac{c_i - p_i}{m_i - 1}$ (that is, the average probability in the chosen branch is at least as high as the average probability of the entire subtree containing all of her choices of branches from v_i). Therefore $\frac{m_j}{c_j} \leq \frac{m_i - 1}{c_i - p_i}$. It follows directly from this that $1 + \frac{1-p_i}{c_j}(m_j) \leq 1 + \frac{1-p_i}{c_i - p_i}(m_i - 1)$. Now we apply the same induction argument as we used in Lemma 5.1:

We claim that $T_i \leq \frac{m_i}{c_i}$ for all $i \in \{1..n\}$ and proceed by induction on m_i. Suppose that this is true for all $m_k < m_i$ and that the cop arrives at vertex v_i. Again, if $p_i \geq \frac{c_i}{m_i}$, then remaining on v_i gives $T_i \leq \frac{m_i}{c_i}$. Otherwise we have $pi < \frac{c_i}{m_i}$ and the cop moves forward, choosing a vertex v_j such that $\frac{m_j}{c_j} \leq \frac{m_i - 1}{c_i - p_i}$. Then with probability p_i, the gambler is captured in one move and with probability $1-p_i$, in $1+\frac{m_j}{c_j}$ moves. So we have $T_i \leq 1 + (1 - p_i)\frac{m_j}{c_j} \leq 1 + (1 - p_i)\frac{m_i - 1}{c_i - p_i}$, by the above observation. In Lemma 5.2, we showed that if $p_i < \frac{c_i}{m_i}$, then $1 + (1 - p_i)\frac{m_i - 1}{c_i - p_i} < \frac{m_i}{c_i}$. Consequently, we have that $T_i \leq \frac{m_i}{c_i}$, as desired. □

5.3 Cop vs. gambler on a general graph

Lemma 5.3. *The gambler can guarantee that the game takes at least n moves on average on any connected graph on n vertices.*

Proof. We first check that the gambler can ensure time at least n on K_n. If the gambler has strategy $p_i = 1/n$ for all $i \in \{1..n\}$, then any strategy of the cop gives probability $1/n$ of capture at each step, and therefore expected capture time n.

K_n is the "easiest" graph for the cop, in the following sense. Suppose for contradiction that there is a graph G which gives expected time $T < n$ for the cop using strategy σ against the uniform gambler strategy; clearly $G \subseteq K_n - xy$ for some $xy \in E(K_n)$. But then σ performed on K_n would give the same expected time T, since the gambler's strategy remains unaffected by a change in the graph, and the cop can play as if on G (since that is a subgraph of K_n). Then the expected capture time on K_n is less than n, which we know to be a contradiction.

Consequently, the gambler can guarantee expected capture time at least n on any size n connected graph. □

Theorem 5.1. *The expected capture time for the cop vs. gambler game on any connected size n graph is n.*

Proof. Let G be any connected graph of size n and let H be a spanning subtree of G. By Lemma 5.2, the cop can capture the gambler in expected time at most n on H, and consequently on G. By Lemma 5.3, we know that the expected capture time is also at least n. □

Note that n remains the value of this game whether things are as good or bad as possible for the cop: she cannot beat n even if she is allowed to pick her starting position after the gambler has chosen (and made public) his strategy, and the gambler cannot beat n even if the cop's initial position is fixed ahead of time. We state this as a lemma and prove it.

Lemma 5.4. *The value of the cop vs. gambler game is n in all three of the following cases:*

(a) The cop's initial position is chosen for her ahead of time.

(b) The cop chooses her own initial position, but before she knows the gambler's strategy.

(c) *The cop chooses her initial position after the gambler makes his strategy known.*

Proof. Note that the first situation is the worst for the cop (as she has no control over her starting point) and the third situation is the best for the cop (as she has as much information as is possible before the start of the game). Therefore it suffices to show that the capture time is at most n in situation (a) and at least n in situation (c).

(a) In this situation, the cop can still get expected capture time n on a tree by Lemma 5.2.

(c) In this situation, the expected capture time is still at most n on K_n by Lemma 5.3.

□

5.4 Cop vs. unknown gambler

In the proof of Lemma 5.2, the cop relied fairly heavily on knowing the gambler's strategy. What if this strategy were not known to the cop? We will (predictably) call the adversary in this variant of the game the "unknown gambler" and will (equally predictably) proceed to consider the expected capture time in this case.

The first question we would like to lay to rest is whether knowing that the adversary is constrained to some time-independent probability distribution (we will henceforth refer to a strategy of the unknown gambler as a **gamble**) is helpful to the cop (even if she does not know what this distribution is). In [1, 6], the "hunter vs. rabbit" scenario is discussed: in this version, the hunter (who fills the role of the cop) remains constrained to the graph and the rabbit is wholly unconstrained—he is free to do as he likes at any turn. On C_n (the n-cycle), the rabbit can guarantee an expected capture time of at least $n \log n$. We will show that the unknown gambler cannot do nearly so well.

Lemma 5.5. *The unknown gambler is strictly weaker than the rabbit.*

Proof. Consider the game played on C_n and label the vertex set $V(C_n)=\{v_1, v_2, \ldots, v_n\}$. Consider the following strategy, Ξ, for the cop: she begins on v_1 and flips a coin to determine her orientation around the cycle, which she then keeps until her adversary is caught. That is, at time t, she is at vertex v_t with probability $1/2$ and at vertex v_{n+2-t} with probability $1/2$ (with subscripts taken modulo n). Suppose that the gamble is defined by $\mathbb{P}(R_t = v_i) = p_i$ for all $i \in \{1..n\}$ and all times $t \in \mathbb{N}$.

We will call each traversal of the cycle by the cop (starting at time t—where $t \equiv 1$ mod n—at v_1 and ending either at v_2 or v_n at time $t + n - 1$) a **trip**. Then note that the probability of failing to capture her quarry on a given trip is $\prod_{k=1}^{n}(1 - p_k) \leq (1 - 1/n)^n \to 1/e$:

> $\prod_{i=1}^{n} x_i \leq \left(\frac{s}{n}\right)^n$ for any $x_1, \ldots, x_n$ with $\sum_{i=1}^{n} x_i = s$. We can see this with the following simple argument. Fix all but two of the terms: say, x_i and x_j. Suppose that $x_i + x_j = a$ for some $a \in [0, 1]$. We maximize the product $x_i x_j$ by noting $x_i x_j = x_i(a - x_i) = a x_i - x_i^2 = \frac{a^2}{4} - (\frac{a}{2} - x_i)^2 \leq \frac{a^2}{4}$ with equality if and only if $x_i = \frac{a}{2}$ and consequently $x_j = a - \frac{a}{2} = \frac{a}{2}$. Since the choice of i and j is arbitrary and $x_i = x_j$, it follows that $x_1 = x_2 = \cdots = x_n = \frac{s}{n}$.

Since each trip is a success (i.e. the unknown gambler is captured) or a failure (the unknown gambler is not captured) independently of any previous trips, the expected number of trips necessary is at most $\dfrac{1}{1 - 1/e} = \dfrac{e}{e-1}$.

Given that the unknown gambler is captured in a given trip, let X be a random variable counting the number of capture points during this trip (that is, we consider the sequence $c_1, c_2, \ldots, c_n$ of cop moves and the sequence $r_1, r_2, \ldots, r_n$ of gambler moves and call any time $k \in \{1..n\}$ with $c_k = r_k$ a **capture point**). Note that $\mathbb{P}(X{>}0) = 1$ by assumption. Let Y be the first capture point. Note that if k is a capture point when the cop goes clockwise from v_1 (which occurs with probability 1/2) then $n+2-k$ is a capture point when the cop goes counterclockwise (which also occurs with probability 1/2). Therefore $\mathbb{E}[Y|X = k] \leq n/2+1$ with equality holding if and only if $k = 1$:

> Let j be a capture point with probability q_j. Then $q_j = q_{n+2-j}$. The average over all capture point locations is then
>
> $$\sum_{j=1}^{n} q_j(1/2 * j + 1/2 * (n + 2 - j)) = \frac{n+2}{2} \sum_{j=1}^{n} q_j = n/2 + 1$$
>
> Therefore if there is exactly one capture point, then its expected location is $n/2 + 1$. If there are $k > 1$ capture points, they occur at times $t_1{<}t_2{<}\ldots{<}t_k$ such that $\frac{1}{k}\sum_{i=1}^{k} t_i = n/2 + 1$, so $t_1 < n/2 + 1$.

Note that $\mathbb{P}(X > 1) > 0$ in any gamble in which $p_i < 1$ for all $i \in \{1..n\}$. If $p_i = 1$ for some $i \in \{1..n\}$, this gamble corresponds to a strategy of a sitter, which we have shown (in Chapter 3) gives expected capture time n. Otherwise, we have the expected capture time is less than

$$\frac{n}{1-1/e} - (1 - n/2) = \left(\frac{e+1}{2(e-1)}\right) n \approx 1.08n < n \log n.$$

□

Lemma 5.6. *The unknown gambler is strictly stronger than the known gambler against the cop doing strategy* Ξ*.*

Proof. Suppose that the gamble on C_n —unbeknownst to the cop—is gamble Π, defined as follows. The (unknown) gambler chooses a set X of $k = \lfloor\sqrt{n}\rfloor$ consecutive vertices uniformly and is equally likely to be at any vertex $v \in X$ at any moment in time (i.e. he is at $v \in X$ at time t with probability $1/k$). Note that since X was chosen uniformly, Ξ is now equivalent to the strategy where the cop chooses a uniformly random starting position and a uniformly random direction. Given that the cop starts outside of X, call her conditional expected capture time Y. Then we have that with probability $p = \left(1 - \frac{1}{k}\right)^k$, she takes time n to get back to her initial position, without having captured the gambler, and with probability $1{-}p$, she takes time $\frac{n-k}{2}$ to enter X and then time k to capture the gambler in X. Therefore we have

$$Y = p(n+Y) + (1{-}p)\frac{n-k}{2} + k$$

and so

$$Y \approx \frac{n}{e-1} + \frac{n+k}{2} \approx 1.082n - .5k$$

The cop starts outside of X with probability $1{-}\frac{1}{k}$, and so the expected capture time is at least

$$\left(1 - \frac{1}{k}\right)Y = 1.082n - 0.582\sqrt{n} - .5$$

which is greater than n for all $n \geq 62$. □

In fact, we now show Ξ is an optimal strategy for the cop against Π, and so the unknown gambler bests any cop on C_n.

Lemma 5.7. *The strategy* Ξ *is an optimal strategy for the cop against the unknown gambler with gamble* Π*.*

Proof. For all times $t < n$, let p_t be the probability that the unknown gambler doing strategy Π is caught by time t, and let $p_{t|>s}$ be the probability that he is caught by time t given that he is not caught by time s. Then note that for all t,

$$p_t = p_{t-1} + (1 - p_{t-1}) p_{t|>(t-1)} \tag{5.1}$$

We will use Equation (5.1) to show that for all $t<n$, p_t is maximized by following the strategy prescribed by Ξ (that is, going to a previously unvisited vertex) at time t. Note that Ξ is optimal if the cop that uses this strategy gets an expected capture time that is less than or equal to the expected capture time she could obtain by using any other strategy. Expected capture time is minimized by maximizing p_t for all t, since

$$\begin{aligned} \mathbb{E}[X] &= \sum_{t=1}^{\infty} t\,\mathbb{P}(X = t) \\ &= \sum_{t=1}^{\infty} \mathbb{P}(X \geq t) \\ &= \sum_{t=1}^{\infty} (1 - \mathbb{P}(X \leq t-1)) \\ &= \sum_{t=1}^{\infty} (1 - p_{t-1}) \end{aligned}$$

(where X is the capture time). When we prove that p_t is maximized for all $t < n$ by moving to a previously unvisited vertex, we show that during the first n steps, the cop is minimizing the expected capture time by moving to a previously unvisited vertex (which is consistent with Ξ). The first n steps then complete one trip around the cycle, and for all $t \geq n$, the optimal moves are the same as at time $t-n$ by the independence of the trips. We now proceed with the proof.

For the sake of labeling convenience, we assume without loss of generality that a cop following strategy Ξ will first visit vertex v_i at time i. Let r_t be the robber's position at time t. Finally, we say the gambler's **index** is $g = i$ if the interval X that he has chosen contains the vertices $v_i, v_i + 1, \ldots, v_{i+k-1}$ (where the indices are always taken modulo n).

Proceed by induction on t. The cop starts, without loss of generality, at vertex v_1. Remaining at vertex v_1 at time 2 yields

$$\begin{aligned} p_2 &= p_1 + (1-p_1)p(r_2 = v_1 | r_1 \neq v_1) \\ &= 1/n + (1-1/n)\frac{1}{k}\sum_{i=n-k+1}^{n+1} \mathbb{P}(g = i | r_1 \neq v_1) \end{aligned}$$

and for each i,

$$\begin{aligned} \mathbb{P}(g = i | r_1 \neq v_1) &= \frac{\mathbb{P}(r_1 \neq 1 | g = i)\mathbb{P}(g = i)}{\mathbb{P}(r_1 \neq v_1)} \\ &= \frac{(1-1/k)1/n}{1-1/n} \\ &= \frac{k-1}{k(n-1)} \end{aligned}$$

However, moving to vertex v_2 at time 2 yields

$$\begin{aligned} p_2 &= p_1 + (1-p_1)p(r_2 = v_2 | r_1 \neq v_1) \\ &= 1/n + (1-1/n)\frac{1}{k}\sum_{i=n-k+2}^{n+2} \mathbb{P}(g = i | r_1 \neq v_1) \end{aligned}$$

$$
\begin{aligned}
&= 1/n + (1 - 1/n)\left(\frac{1}{k}\sum_{i=n-k+2}^{n+1} \mathbb{P}(g = i | r_1 \neq v_1) + \mathbb{P}(g = 2 | r_1 \neq v_1)\right) \\
&= 1/n + (1 - 1/n)\left(\frac{k-1}{k(n-1)} + \mathbb{P}(g = 2 | r_1 \neq v_1)\right)
\end{aligned}
$$

where

$$
\begin{aligned}
\mathbb{P}(g = 2 | r_1 \neq v_1) &= \frac{\mathbb{P}(r_1 \neq v_1 | g = 2)\mathbb{P}(g = 2)}{\mathbb{P}(r_1 \neq v_1)} \\
&= \frac{1/n}{1 - 1/n} = \frac{1}{n-1}
\end{aligned}
$$

Therefore p_2 is maximized by taking a step to an unvisited vertex.

Now fix $t<n$ and suppose that the result holds for all $s < t$, that is, p_s is maximized by taking a step to a previously unvisited vertex at time s. By Equation (5.1),

$$
p_t = p_{t-1} + (1 - p_{t-1})p_{t|>(t-1)}
$$

is maximized by a step to an unvisited vertex at time t if this maximizes $p_{t|>(t-1)}$ (since p_{t-1} is maximized by the inductive hypothesis), by the following argument:

Suppose that $p_{t|>(t-1)} \leq q$. Then

$$
\begin{aligned}
(1 - p_{t-1})p_{t|>(t-1)} &\leq (1 - p_{t-1})q \\
\implies p_{t-1} + (1 - p_{t-1})p_{t|>(t-1)} &\leq p_{t-1} + (1 - p_{t-1})q
\end{aligned}
$$

For each $i < t$, let A_i be the event that the robber was **not** at vertex v_i at time i.

Then if the cop's step at time t is to an unvisited vertex (i.e. to v_t) then

$$p_{t|>(t-1)} = \mathbb{P}(r_t = v_t | A_1 \cap A_2 \cap \cdots \cap A_{t-1})$$

And we have that for all events A,

$$\mathbb{P}(r_t = v_t | A) = \sum_{i=t-k+1}^{t} \frac{1}{k} \mathbb{P}(g = i | A)$$

Let i be any index such that $v_t \in X$ and let j_i be the number of vertices in X with indices from 1 to $t-1$ (i.e. $j_i = |\{i, i+1, \ldots, i+k-1\} \cap \{1, 2, \ldots, t-1\}|$).

$$\begin{aligned}
\mathbb{P}(g = i | A_1 \cap A_2 \cap \cdots \cap A_{t-1}) &= \frac{\mathbb{P}(A_1 \cap A_2 \cap \cdots \cap A_{t-1} | g = i)\mathbb{P}(g = i)}{\mathbb{P}(A_1 \cap A_2 \cap \cdots \cap A_{t-1})} \\
&= \frac{(1 - 1/k)^{j_i} \frac{1}{n}}{\mathbb{P}(A_1 \cap A_2 \cap \cdots \cap A_{t-1})}
\end{aligned}$$

So that

$$\begin{aligned}
p_{t|>(t-1)} &= \mathbb{P}(r_t = v_t | A_1 \cap A_2 \cap \cdots \cap A_{t-1}) \\
&= \frac{1}{k} \sum_{i=t-k+1}^{t} \frac{(1 - 1/k)^{j_i}}{n\mathbb{P}(A_1 \cap A_2 \cap \cdots \cap A_{t-1})}
\end{aligned}$$

If the cop's step at time t is to vertex v_{t-1} then we have

$$\begin{aligned}
p_{t|>(t-1)} &= \mathbb{P}(r_t = v_{t-1} | A_1 \cap A_2 \cap \cdots \cap A_{t-1}) \\
&= \frac{1}{k} \sum_{i=t-k}^{t-1} \frac{(1 - 1/k)^{j_i}}{n\mathbb{P}(A_1 \cap A_2 \cap \cdots \cap A_{t-1})}
\end{aligned}$$

And finally if the cop moves to v_{t-2} at time t then

$$\begin{aligned} p_{t|>(t-1)} &= \mathbb{P}(r_t = v_{t-2}|A_1 \cap A_2 \cap \cdots \cap A_{t-1}) \\ &= \frac{1}{k} \sum_{i=t-k-1}^{t-2} \frac{(1-1/k)^{j_i}}{n\mathbb{P}(A_1 \cap A_2 \cap \cdots \cap A_{t-1})} \end{aligned}$$

Therefore letting $S = \frac{1}{k} \sum_{i=t-k+1}^{t-2} \frac{(1-1/k)^{j_i}}{n\mathbb{P}(A_1 \cap A_2 \cap \cdots \cap A_{t-1})}$ we have

$$\begin{aligned} \mathbb{P}(r_t = v_t|A_1 \cap A_2 \cap \cdots \cap A_{t-1}) &= S + \frac{(1-1/k)^{j_{t-1}} + (1-1/k)^{j_t}}{kn\mathbb{P}(A_1 \cap A_2 \cap \cdots \cap A_{t-1})} \\ &= S + \frac{(1-1/k) + 1}{kn\mathbb{P}(A_1 \cap A_2 \cap \cdots \cap A_{t-1})} \\ &= S + \frac{2-1/k}{kn\mathbb{P}(A_1 \cap A_2 \cap \cdots \cap A_{t-1})} \end{aligned}$$

while

$$\begin{aligned} \mathbb{P}(r_t = v_{t-1}|A_1 \cap A_2 \cap \cdots \cap A_{t-1}) &= S + \frac{(1-1/k)^{j_{t-k}} + (1-1/k)^{j_{t-1}}}{kn\mathbb{P}(A_1 \cap A_2 \cap \cdots \cap A_{t-1})} \\ &\leq S + \frac{(1-1/k) + (1-1/k)}{kn\mathbb{P}(A_1 \cap A_2 \cap \cdots \cap A_{t-1})} \\ &= S + \frac{2-2/k}{kn\mathbb{P}(A_1 \cap A_2 \cap \cdots \cap A_{t-1})} \end{aligned}$$

and

$$\begin{aligned} \mathbb{P}(r_t = v_{t-2}|A_1 \cap A_2 \cap \cdots \cap A_{t-1}) &= S + \frac{(1-1/k)^{j_{t-k-1}} + (1-1/k)^{j_{t-k}}}{kn\mathbb{P}(A_1 \cap A_2 \cap \cdots \cap A_{t-1})} \\ &< S + \frac{2-2/k}{kn\mathbb{P}(A_1 \cap A_2 \cap \cdots \cap A_{t-1})} \end{aligned}$$

Consequently, $p_{t|>(t-1)}$ is maximized by going to v_t at time t, and so p_t is maximized by going to v_t at time t, as desired. □

Therefore we have the following as an immediate corollary of Lemmas 5.5, 5.6, and 5.7.

Theorem 5.2. *The unknown gambler is strictly stronger than the known gambler and strictly weaker than the rabbit.*

Chapter 6

Capture Time in (Traditional) Cops & Robbers

In this chapter we discuss the question of capture time in the original variation of cops and robbers. That is, the cop and robber are moving alternately and with full information on a finite, connected, **directed** graph G on n vertices. We begin with an upper bound.

Lemma 6.1. *On any connected cop-win graph on n vertices, the original cop and robber game takes at most n^2 steps.*

Proof. Let $c(t)$ and $r(t)$ be the cop's and robber's (respectively) positions at time t. Consider the set $V(G) \times V(G)$ of possible ordered pairs of positions $(c(t), r(t))$ for all times t. We call such a pair a **state**. If at some time t, the players are in state $(c(t), r(t))$ and there is a series of cop moves $c(t+1), c(t+2), \ldots, c(t+k)$ for some k which forces the robber to be caught by time $t+k$, then we call $(c(t), r(t))$ a **winning state**.

Let $(c(t), r(t))$ be a winning state for some t and let t be the first time at which this state occurs. Then if $(c(t+1), r(t+1)), \ldots, (c(t+k), r(t+k))$ is a series of moves (with optimal play by both parties) which ends with the robber's capture, then for all $i \in \{1..k\}$, $(c(t+i), r(t+i)) \neq (c(t), r(t))$:

> Let $(c(t), r(t))$ be a winning state and suppose that $c(t+1), c(t+2), \ldots, c(t+k)$ is a sequence of optimal moves for the cop. Assume for sake of contradiction that $(c(t), r(t)) = (c(t+i), r(t+i))$ for some $i \in \{1..k\}$. Then the sequence of moves $c(t), c(t+i+1), c(t+i+2), \ldots, c(t+k)$ is a winning sequence for the cop, and is shorter than the previously mentioned allegedly optimal sequence.

Since there are only $|V(G) \times V(G)|$ possible states and an optimal cop strategy on a cop-win graph does not allow for repetition of states, the capture time cannot exceed n^2. □

Note that this bound applies to directed graphs and also to graphs on which the cop and robber may use different edge sets. As will be discussed in the following section, this is a gross overestimate in the case of undirected graphs, where the capture time does not in fact exceed $n-4$ (for all undirected graphs on $n \geq 7$ vertices). However, we shall see that it is not such a bad bound in the case of general directed graphs!

6.1 Undirected Graphs

In the original formulation of the game, the notion of capture time requires a game which is going to end in a finite number of steps in order to be interesting; that is, the number of cops playing must be at least as high as the cop number of the graph. In [8], the authors note that the capture time of any graph G is bounded by $|V(G)|^{c(G)+1}$ where $c(G)$ is the cop number of G. However, this bound is very far from sharp in general. Later, in [26], the length of games for chordal graphs are analyzed. The first concentrated effort on describing capture time in the original variation of the game is by Bonato, Golovach, Hahn, and Kratochvil [11]. In that work (and later in a paper of Gavenčiak [24]) observations are largely restricted to cop-win graphs (that is, graphs with $c(G)=1$). The authors prove that on a cop-win graph with $n \geq 5$ vertices, the cop wins in at most $n-3$ steps. Furthermore, they construct an example of an infinite family of graphs for which $n-4$ steps are needed. Gavenčiak later closes this gap by improving the upper bound to $n-4$ for all $n \geq 7$. Bonato et al. also show that for a large class of graphs, the capture time is bounded by $n/2$. Below, we will explain in brief some of their results.

To show that the capture time is bounded by $n-3$ for all $n \geq 5$, the authors proceed by induction with the base case proven using an exhaustive check of all order 5 cop-win graphs. Suppose now that the statement is true for all order n cop-win graphs and let G be a cop-win graph on $n+1$ vertices. Then G contains a corner u and since any retract of a cop-win graph is cop-win [41], $G-u$ is a cop-win graph of order n. Therefore by the induction hypothesis the capture time on $G-u$ is at most $n-3$. Let v be a vertex of G that dominates the corner u. On G, the cop plays the same strategy as she played on $G-u$, with the exception that when the robber is on u, the cop plays as though he is on v. After $n-3$ moves with this strategy, either the game is over or the robber is on u and the cop is on v, in which case she can capture on her next move. Therefore the capture time on G is at most $n-3+1 = |V(G)|-3$.

To show that $n-4$ is a lower bound on the capture time on an order n graph, Bonato et al. construct an infinite family of graphs $H(n)$ with capture time exactly $n-4$ (with optimal play by both parties). Consider the graph H in Figure 6.1. Note that it is cop-win and has a unique corner (the vertex z).

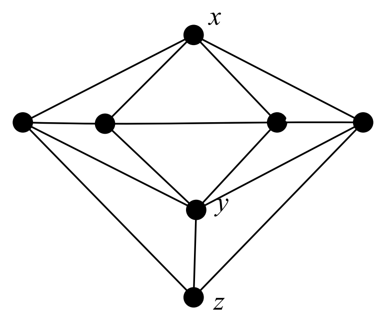

Figure 6.1: H: A cop-win graph with a unique corner

Using H, the authors construct an infinite family of graphs, $H(n)$ for $n \geq 7$ with vertices with labels in $\{1..n\}$ (the integers between 1 and n, inclusive) where vertices with labels in $\{1..7\}$ induce H (so that $x = 5, y = 3, z = 7$) and the remaining vertices on the path joined to x and y are $6, 2, 1, 4$. For the rest of the vertices ($i > 7$), vertex i is joined to $j < i$ if j is $i-4, i-3$, or $i-1$. See Figure 6.2 for $H(11)$.

$H(n)$ has the following properties (see Theorem 4 in [11]).

(a) $H(n)$ is planar.

(b) $H(n)$ is cop-win and has a unique corner (the vertex labeled n).

(c) The capture time on $H(n)$ is exactly $n-4$.

We will leave the verification of the first two items to the reader (see the proof of Theorem 4 in [11]) and focus on the proof of the fact that the capture time on $H(n)$ is at least $n-4$ (that is, we are also leaving the verification of the other direction of the inequality to the reader—see the proof in [11] for an optimal cop strategy). To

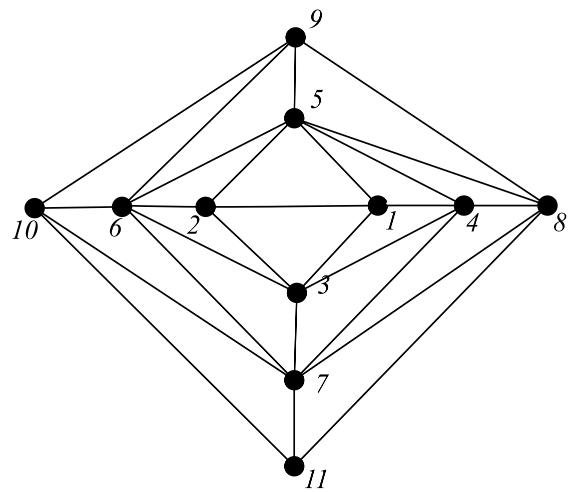

Figure 6.2: $H(11)$

show that the robber has a strategy that guarantees his safety for at least $n-4$ moves, note that the robber may use the lowest index vertex available to him. The robber may begin on any of the vertices in $\{1..6\}$ as no single vertex of $H(n)$ dominates all of these. Let $c(t)$ and $r(t)$ be the cop's and robber's positions, respectively, at time t for all $t \geq 0$. Suppose that $r(t-1) \leq 4$; then the robber can move to $r(t) \leq 7$. Otherwise if $5 \leq r(t-1) \leq n-1$, the robber's options are summarized in Table 6.1. Therefore we have $r(t) \leq 6+t$ for all $t \geq 0$, and since n is the unique corner of $H(n)$, the robber is not caught until $r(t) = n$ (at which the cop can move to a vertex $c(t+1)$ that dominates the corner, and whatever vertex the robber chooses at $t+1$, the cop will capture with $c(t+2) = r(t+1) \in N(n)$) and so will not be caught in fewer than $n-6+2-n-4$ cop moves.

Table 6.1: A table of the robber's moves

$c(t)$	$r(t)$
$i-4$	$i+1$
$i-3$	$i-1$
$i-1$	$i-3$
$i+1$	$i-4$
$i+3$	$i-4$
$i+4$	$i-4$

6.2 Directed Graphs

As we saw in the previous section, a linear bound on capture time exists on all undirected graphs. Can we do so well for more general graphs? It turns out that the answer is in fact no for a general directed graph, as evidenced by the following counterexample.

Lemma 6.2. *There exists an infinite family of directed graphs on n vertices with capture time $\Theta(n^2)$.*

Proof. We define a **ring digraph** $R(k)$ to be a reflexive directed graph on $n = 2k+1$ vertices consisting of:

- an "outer ring" comprised of a (counterclockwise)-directed k-cycle
- an "inner ring" comprised of a (counterclockwise)-directed $(k-1)$-cycle
- arcs from a vertex in the inner ring to a vertex in the outer ring configured such that $k-2$ vertices in the inner ring are incident with one such arc, and 1 vertex in the inner ring is incident with two such arcs

- an “internal vertex” (marked C in Figure 6.3) that is out-directed to every vertex in the inner ring
- an “external vertex” (marked R in Figure 6.3) incident with two arcs as in Figure 6.3

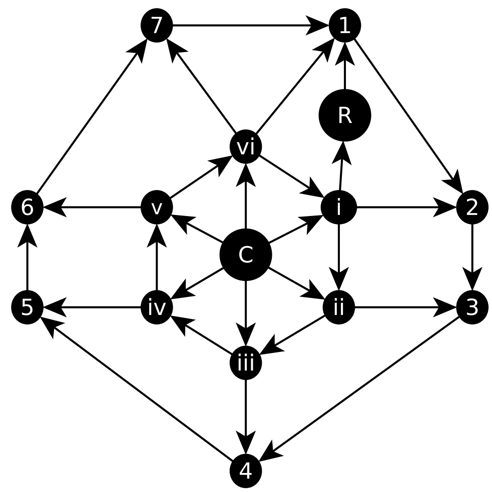

Figure 6.3: The ring digraph $R(7)$

We claim that for all k, $R(k)$ is cop-win and the capture time on $R(k)$ is $\Theta(n^2)$. In particular, the capture time on $R(k)$ is $(n^2 - 4n + 3)/4$, where $n = |V(G)|$.

We describe a strategy where the cop starts on the vertex labeled C in Figure 6.3 and show that with both players moving optimally, the game takes $(k-1)^2 + 1 = k^2 - 2k$ moves from this starting position. Then we show that this is the best starting position for the cop.

Let C_i be the cop's position after the i^{th} move (so C_0 is her starting position), and define R_i analogously. Then if $C_0 = C$ and $R_0 = R$, then the cop must play $C_1 = i$ (the robber can remain on R until the cop is on this vertex). The robber's only choice is to play $R_1 = 1$. Now the cop must move along the inner ring, as the robber can pass whenever the cop passes (and if she moves onto the outer ring, the robber wins). Now the only way that the cop can win is to be on vertex vi in Figure 6.3 (that is, the vertex on the inner ring with outdegree 3) while the robber is on vertex 7. That is, the cop wants to be one step behind the robber. Then the robber can either pass or move onto vertex 1, and will be captured in one move in either case. But after move 1, the cop is one step ahead of (i.e. $k-1$ steps behind) the robber. Therefore it will take $k-1$ full trips (i.e. $(k-1)^2$ moves) to come to this position, as every time the cop makes a full trip around the inner ring, she is one more step ahead of the robber.

If the cop's initial position was anywhere but on vertex C, then the robber could play $R_0 = C$ and remain there for the rest of the game, so $C_0 = C$ is the cop's only option. □

6.3 Graph Pairs

We may try to make the game on a directed graph a bit easier for the cop by introducing edges only she can use (which we will call "cop edges"). In particular, let dashed arcs be traversible by the cop but not by the robber and solid arcs be traversible by both players. It turns out that this is still not enough to make cop-win digraphs have linear capture time.

Theorem 6.1. *Let G be a directed n-cycle with a dashed arc connecting a single vertex to a vertex that is at distance 2 from it in the cycle, as in Figure 6.4. Let G be reflexive—that is, the players are allowed to stay at a vertex at any given time. This*

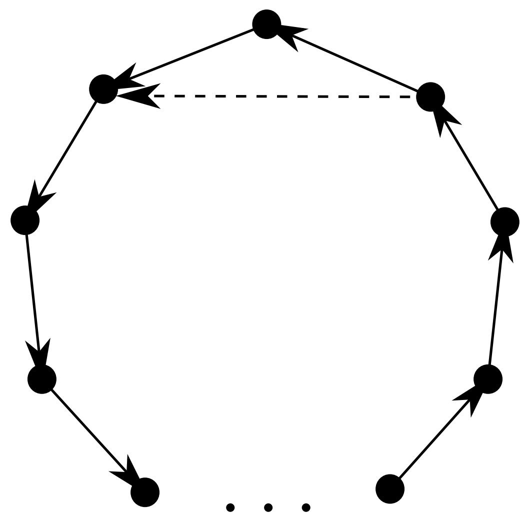

Figure 6.4: A digraph with one cop edge

graph is a cop-win directed graph with capture time $\Theta(n^2)$.

Proof. Suppose that the cop places herself on the vertex with outdegree 2. Then the robber places himself one vertex behind the cop—that is, the cop is now at distance $n-2$ to the robber. After each trip around, with the robber going around the black n-cycle and the cop taking the dashed shortcut, the distance from the cop to the robber is one less than during the previous trip. That means that it takes $n+1$ trips before the robber is at distance 1 from the cop and it is the cop's move. Since each trip takes $n-1$ moves, the total capture time is then $(n+1)(n-1)+1 = n^2$. □

6.4 Future Work

Note that the ring digraph described in Lemma 6.2 is not strongly connected. So it is reasonable to ask whether all strongly connected digraphs have linear capture time.

Note also that the graph in Example 6.1 is strongly connected (though without the dedicated cop edge, it would not be cop-win). It may also be interesting to study whether all strongly connected digraphs with at least one cop-only edge which have quadratic capture time share the property that the black arcs form a robber-win digraph.

Chapter 7

Speculation and Future Directions

7.1 Summary of Results

We will summarize the main results presented in this thesis in the following chart, with the key assumptions of the respective sections emphasized. This chart may give some insight into potential future directions, some of which we will consider in more detail in the following section. Other than where explicitly noted otherwise, games are played on a graph G (assumed to be simple, undirected, and connected) with n vertices.

For each result in the chart (first column), we point out the following parameters (columns two through five):

- **Move type**: simultaneous or alternating
- **Pursuer move constraints**: it is assumed other than where explicitly noted otherwise—with the label "unconstrained"—that the pursuer moves from vertex to adjacent vertex

- **Evader move constraints**: it is assumed other than where explicitly noted otherwise—with the label "unconstrained"—that the evader moves from vertex to adjacent vertex

- **Visibility**: Full (the players see each other at all times), none (the players never see each other until capture occurs), or somewhere in between

Result	Move type	Pursuer moves	Evader moves	Visibility
Let $x_0 \in V(G)$ be any vertex in G. Let $\{x_0, x_1, x_2, \dots\}$ be any random walk on G beginning at x_0. Then $\mathbb{P}(d(x_0, x_4) < 4) \geq \frac{1}{4}n^{-2/3}$.	N/A	N/A	N/A	N/A
A cop will capture a drunk on G in expected time $n+o(n)$.	Alt.	Must move	Must move; random	Full
A cop will capture a sitter in expected time $n-1$ on a tree; furthermore, the strategies defined in Section 3.1 are optimal strategies for the players (and the sitter strategy is the unique optimal one).	N/A	None	Immobile	None
The expected capture time in the cop vs. sitter game is between $\frac{n+1}{2}$ and $n-1$ on G, and is strictly less than $n-1$ if G is not a tree.	N/A	None	Immobile	None

The DFP strategy defined in Section 3.1 is asymptotically optimal on K_n for the cop. It is also an optimal response for the robber.	Simul.	None	None	None
The expected capture time for a cop doing a uniform depth first pursuit on a binary tree is greater than $\frac{(n-15)(n-1)}{8}$.	Alt.	None	None	None
A graph is hunter-win (i.e. mole is guaranteed to be captured in bounded time) if and only if it is a lobster.	Simul.	Unconstrained	Must move	Vis. hunter; invis. mole
A graph is hunter-win if and only if it satisfies the constraints described in Theorem 4.3.	Simul.	Unconstrained	None	Vis. hunter; invis. mole
The expected capture time of the cop vs. gambler game is n on G; this is true regardless of whether the cop gets to choose her starting position.	Simul.	None	Fixed probability distribution	Cop knows gambler's distribution
The unknown gambler is strictly stronger than the known gambler and strictly weaker than the rabbit [1].	Simul.	None	Fixed probability distribution	None
There exists an infinite family of directed cop-win graphs with expected capture time $\Theta(n^2)$. There also exists an infinite family of "graph pairs" (see section 6.3).	Alt.	None	None	Full

7.2 Further Investigations into Capture Time

7.2.1 Cop vs. Visible Robber

In Chapter 2, we saw that in the game of a cop playing with full information against a random walker, a greedy distance minimizing strategy can lead the cop astray. Recall the confounding counterexample: the "ladder to the basement" graph of Figure 7.1.

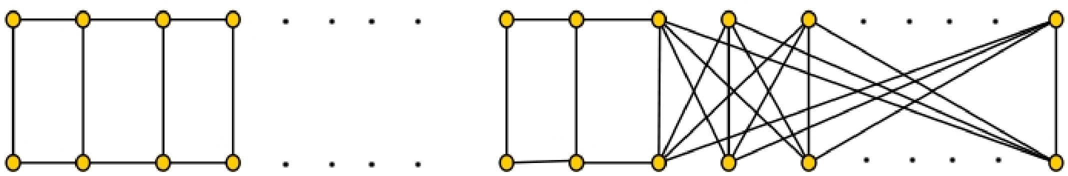

Figure 7.1: The Ladder to the Basement

However, this was a cop who used insistently poor decision-making to break ties: when presented with at least two distance-minimizing options at a given step, she was forced to choose a vertex that kept her from making progress toward the capture of her quarry. This leaves open the question of whether greedily minimizing distance would actually suffice to give capture time $n+o(n)$ if such ties are broken by a random choice. In the case of the graph from Figure 7.1, the random-tie-breaking greedy cop does not have a problem capturing the drunk in fewer than n steps.

Another question we may like to address in the case of a cop chasing a random walker is that of the worst case graph. We conjecture that this graph contains a clique of size $cn^{1/3}$ for some constant $c \in \mathbb{R}$ with a path of length $n - cn^{1/3}$ attached at one vertex (that is, a lollipop graph).

Extending our interest to more intelligent varieties of the visible robber, we refer back to the main question posed in Chapter 6. Does there exist a strongly connected directed graph with a superlinear capture time?

7.2.2 Cop vs. Invisible Robber

In the discussion in Chapter 2, we consider the game of cop vs. drunk played with full information. The cop's strategy relies, on a general graph, on knowing the drunk's position at various times, in order to be able to continue heading in the right direction. What about the case of a cop and drunk playing in the dark? Certainly this does not impact the random walker (who was not using any information about the cop anyway), but it may make things more difficult for the cop. How much more difficult, exactly? Can she still capture in linear time? Or perhaps even in time $n + o(n)$, as in the full information case?

Consider the "extended barbell graph," $B_{n,k}$, which we will define as follows. $B_{n,k}$ consists of a path of length $n - 2k$ connecting two copies of K_k, where $|V(B_{n,k})| = n$.

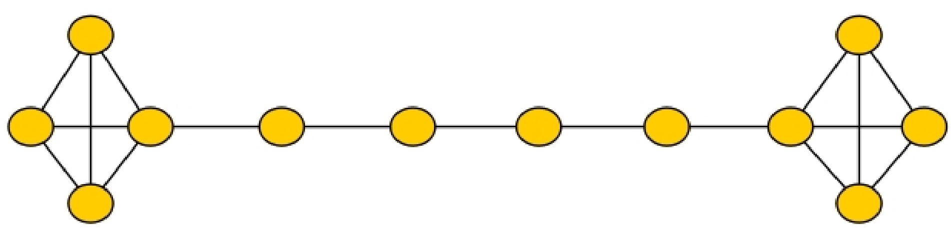

Figure 7.2: The graph $B_{12,4}$

Suppose that the drunk starts in one of the cliques, and the cop starts in the middle of the path. Since the drunk is invisible, the cop can choose one of the cliques with probability 1/2 and head directly toward it (taking time $\frac{n-2k}{2}$ to get there). Let

$c > 0$ and consider the following strategy: the cop chooses a random clique and goes there immediately; then she spends time ck in the clique before heading directly for the other clique and spending time ck there, and so on until the drunk is captured. Consider the graph $B_{n,0.01n}$.

With probability 1/2, the cop chooses the clique occupied by the drunk. For $k \in \mathbb{N}$, let $p(k)$ be the probability that the drunk remains in the same clique for k steps, so after the cop spends time $0.49n$ getting to the chosen clique, the drunk is still in that clique with probability $p(.49n)$ when the cop arrives – i.e. the cop does not manage to capture the drunk on her way over to the clique. Then with probability e^{-c}, she fails to capture the drunk in time $0.01cn$ and spends another $.98n + .01cn$ heading to the other clique and searching there, and then heading all the way back to the original clique, where the drunk is still milling about with probability $p(.01cn + .98n + .01cn + .98n) = p(1.96n + .02cn)$. Supposing that the cop always captures by the second visit, the capture then takes another $.01n$ expected moves.

On the other hand, with probability 1/2, the cop heads first toward the wrong clique; then she spends time $.49n + .01cn$ in the wrong clique before heading to the correct one for another $.98n$ moves. The drunk is still in the original clique with probability $p(1.47n + .01cn)$, and then the same process repeats as in the first case.

Therefore the expected capture time, T, satisfies the following inequality:

$$T \geq 1/2[p(.49n)(.49n + m)] + 1/2[.49n + .01cn + p(1.47n + .01cn)(.98n + m)]$$

where

$$\begin{aligned} m &= e^{-c}(.01cn + .98n + .01cn + p(1.96n + .02cn)(.98n + .01n)) \\ &= (.02c + .98 + .99p(1.96n + .02cn))e^{-c}n \end{aligned}$$

We have $p(k) \geq 1 - \frac{k}{2}\left(\frac{100}{n}\right)^2$ for all k, so

$$\begin{aligned} m &\geq \left(.02c + .98 + .99\left(1 - \left(\frac{1.96 + .02c}{2}\right)\left(\frac{100^2}{n}\right)\right)\right)e^{-c}n \\ &= (1.97 + .02c)e^{-c}n - 50(196 + 2c)e^{-c} \end{aligned}$$

Therefore we have

$$T \geq \frac{1}{2}(1.96 + .01c + e^{-c}(3.92 + .04c))n - (4201.75 + 24.5c + e^{-c}(19452 + 247.25c - .5c^2)) + O(1/n)$$

The coefficient on n is always greater than 1 for all values of c (its minimum value is approximately 1.015, achieved when c is approximately 6.02.). Therefore, (for large n) on the graph $B_{n,0.01n}$ the cop cannot capture the drunk in expected time n if she spends time $c(.01n)$ searching for the drunk in the cliques, for any $c > 0$.

Little is known about the case of a cop playing against a general invisible robber. How can we characterize cop-win graphs in this game, and what is the worst-case expected capture time on a cop-win graph against an invisible adversary?

7.3 Patrolling Schemes

In this section we develop the concept of a patrolling scheme. This is a natural concept that arises from application: what is the best “beat” for a cop to patrol in her assigned network if she wishes to prevent a robber from perpetrating his heinous crime at one of several enticing locations? There is some literature in the field of computational geometry that considers this question [46] as well as some study in operations research [32]. However, we suggest a probabilistic approach that assumes an expert adversary: one with enough resources to have an eye on the cop at all times, and consequently require a mixed strategy in order to conquer.

We want to establish a general mathematical notion of a patrolling scheme for a cop on any (locally finite) graph G. This means that we are looking to define rigorously the “danger” of a particular situation – that is, how dangerous it is for the robber to commence a robbery at a certain point along the cop’s beat. Then the cop’s goal is to choose a patrolling scheme that, in some sense, maximizes over all possible situations the minimal danger of a situation. First we’ll consider an example of a particular patrolling scheme on C_4, and then we’ll formalize and generalize this notion. We will finish here with another example (computing an optimal patrolling scheme for K_3) and leave it an open problem to develop this concept further and compute optimal patrolling schemes on various classes of graphs.

7.3.1 Example: α-momentum cop on C_4

Define an *α-momentum cop* on a cycle C_n to be a cop assigned to patrol the graph, starting in some direction, and changing direction at each vertex with probability $1 - \alpha$. Now consider an α-momentum cop patrolling the cycle C_4. Suppose that, unbeknownst to the cop, there is a robber who knows the cop's current position, patrolling history, and her patrolling scheme. Suppose further that he is interested in committing a felony on vertex 0 (see Figure 7.3).

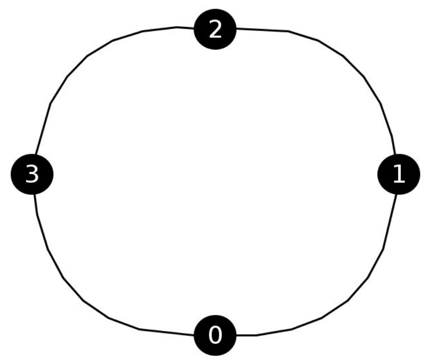

Figure 7.3: The cycle C_4.

Then there are three situations to consider:

(a) The cop is on vertex 1 and heading clockwise (this is the same as the cop being on vertex 3 and heading counterclockwise).

(b) The cop is on vertex 2 (and heading in either direction).

(c) The cop is on vertex 3 and heading clockwise (this is the same as the cop being on vertex 1 and heading counterclockwise).

Then letting $p_i(t)$ be the probability that from situation i, the α-momentum cop will be at vertex 0 in (exactly) t steps, we have the following equations:

$$p_1(2k+1) = \alpha(1-\alpha)^k \text{ for all } k \geq 0 \quad (7.1)$$

$$p_2(2k) = \alpha(1-\alpha)^{k-1} \text{ for all } k \geq 1 \tag{7.2}$$

$$p_3(2k+1) = \alpha^2(1-\alpha)^{k-1} \text{ for all } k \geq 1, p_3(1) = 1-\alpha \tag{7.3}$$

For a fixed situation i, define the *danger function* $W_i(\alpha) = \sum_{k=0}^{\infty} \frac{p_i(k)}{k}$. Note that under this definition, the danger functions are unit-invariant. That is, changing the units of time with which we are dealing would simply change the "danger index" by a constant.

Now working out the danger index in each of the three situations, equation (1) gives us

$$\begin{aligned} W_1(\alpha) &= \sum_{k=0}^{\infty} \frac{\alpha(1-\alpha)^k}{2k+1} \\ &= \frac{\alpha}{2\sqrt{1-\alpha}}\, ln\left(\frac{1+\sqrt{1-\alpha}}{1-\sqrt{1-\alpha}}\right) \end{aligned}$$

Equation (2) yields

$$\begin{aligned} W_2(\alpha) &= \sum_{k=1}^{\infty} \frac{\alpha(1-\alpha)^{k-1}}{2k} \\ &= \frac{\alpha}{2(\alpha-1)}\, ln(\alpha) \end{aligned}$$

And finally, equation (3) says

$$
\begin{aligned}
W_3(\alpha) &= 1-\alpha+\sum_{k=1}^{\infty}\frac{\alpha^2(1-\alpha)^{k-1}}{2k+1}\\
&= \frac{\sqrt{1-\alpha}}{2}\,ln\left(\frac{1+\sqrt{1-\alpha}}{1-\sqrt{1-\alpha}}\right)-\frac{1}{\sqrt{1-\alpha}}\,ln\left(\frac{1+\sqrt{1-\alpha}}{1-\sqrt{1-\alpha}}\right)\\
&\quad+\frac{1}{2(1-\alpha)^{3/2}}\,ln\left(\frac{1+\sqrt{1-\alpha}}{1-\sqrt{1-\alpha}}\right)-\frac{1}{1-\alpha}+2
\end{aligned}
$$

This tells us that, as expected, when the cop and robber are in situation 1 or 2, the deterministic case is best for the cop (in that it maximizes the danger for the robber). However, when they are in situation 3, the lower the α the better – that is, of course, the cop would maximize the danger if she had a higher probability of turning around at that point.

The three danger functions are graphed together in Figure 7.4. The graphs of $W_2(\alpha)$, in green, and $W_3(\alpha)$, in blue, intersect at the point $\alpha = 0.79648$, at which point their common value is 0.4453. This tells us that in order to maximize the minimal danger, the cop should choose a patrolling scheme on C_4 with a probability of 0.20352 of turning around.

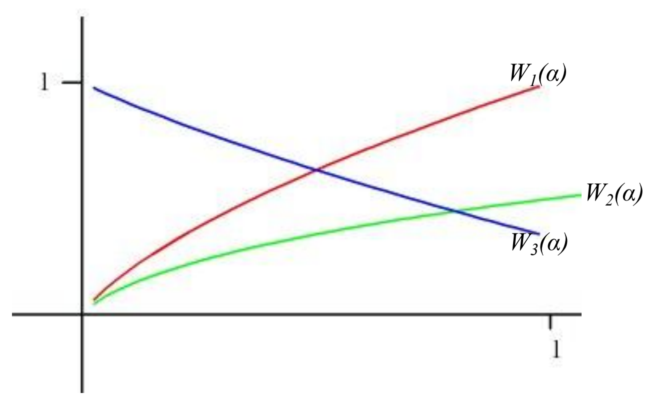

Figure 7.4: Graph of the functions $W_1(\alpha)$, $W_2(\alpha)$, and $W_3(\alpha)$.

7.3.2 Topology on the space of legal walks

To generalize this notion of a patrolling scheme to any (locally finite) graph, we define the following space, and observe some of its properties. Fix a locally finite graph G.

Definition 7.1. Define a **legal walk** on a graph G to be an infinite walk taken on the graph. In the case of a (locally finite) graph G, we fix an initial vertex v_0 and define a legal walk as any infinite walk taken on the graph that starts at v_0.

Let $W(G)_\infty$ (or simply W_∞ when this is not confusing) be the space of all infinite legal walks, with the topology defined by the base sets $B(\sigma)$, where for any finite walk on σ on G, $B(\sigma)$ is the set of all legal walks on G that begin with σ. Give W_∞ the metric defined by the function $d(\sigma, \sigma') = 2^{-d}$ where d is the position of the first disagreement between the legal walks σ and σ'. Then the metric space W_∞ satisfies the following properties.

Theorem 7.1. *W_∞ is complete.*

Proof. Suppose that $\{S_k\}_{k=1}^{\infty}$ is a Cauchy sequence of legal walks on G. We want to show that it converges to a point in W_∞. Fix $d > 0$, and let $M \in \mathbb{N}$ such that for all $m, n > M$, $d(S_m, S_n) < 2^{-d}$. Then for all $m, n > M$, the legal walks S_m and S_n agree in the first $d - 1$ terms, so the limit σ of the sequence must agree in the first $d - 1$ terms. Building σ in this way as d tends to ∞ tells us that at every (finite) position, σ agrees with some legal walk. Therefore, σ must be a legal walk. □

Theorem 7.2. *W_∞ is compact.*

Proof. Let $\{S_k\}_{k=1}^{n}$ be a sequence of legal walks on a (locally finite) graph G with $V(G) = \{v_0, v_1, v_2, \ldots\}$. Determine a subsequence with limit point σ in the following manner. Let $\sigma|_d$ be the vertex in position d of the sequence σ of vertices. Choose $\sigma|_0 = v_i$ where

$$i = \min\{j : \text{ there exist an infinite number of } S_k \text{ with } S_k|_0 = v_j\}$$

and remove all S_k that disagree with σ in position 0. Of the walks that remain, grant immunity to one of them, meaning that this walk will remain in our subsequence.

Now for all $d > 0$, choose $\sigma|_d = v_i$ where

$$i = \min\{j : v_j \text{ is adjacent to } S|_{d-1} \text{ and there exist an infinite number of } S_k \text{ with } S_k|_d = v_j\}$$

and again remove all S_k that disagree with σ in position d which have not been granted immunity.

The walks that remain form a subsequence converging to σ. Note that constructing σ in this manner ensures that σ is itself a legal walk, since for all d, there are an infinite number of legal walks that agree with σ in the first d positions. □

7.3.3 Patrolling schemes

Definition 7.2. A **patrolling scheme** is a function associating to each finite path $v_0, v_1, \ldots, v_{n+1}$ a number $P(v_{n+1} \mid v_0, v_1, \ldots, v_n)$ with the property that $\sum_u P(u \mid v_0, v_1, \ldots, v_n) = 1$, where the sum is taken over all u adjacent to v_n.

Definition 7.3. A sequence $\{\mu_n\}_n$ of probability measures on the spaces $\{W_n\}_n$ of length-n walks beginning at v_0 is **compatible** if for all walks $v_0, \ldots, v_{n-1}$,

$$\mu_{n-1}(v_0, \ldots, v_{n-1}) = \sum_u \mu_n(v_0, \ldots, v_{n-1}, u).$$

Definition 7.4. To any patrolling scheme P we associate a compatible sequence $\{\mu_n\}_n$ of probability measures on the spaces $\{W_n\}_n$ of length-n walks beginning at v_0 by defining

$$\mu_n(v_0, \ldots, v_n) = \prod_{k=1}^{n} P(v_k \mid v_0, \ldots, v_{k-1})$$

for all walks $v_0, \ldots, v_n \in W_n$. This sequence is called the **associated sequence of probability measures** for P.

Theorem 7.3. *For each compatible sequence of probability measures $\{\mu_n\}$ on W_n, there exists a unique patrolling scheme P for which $\{\mu_n\}_n$ is the associated sequence of probability measures.*

Proof. Suppose first that there is a patrolling scheme P on G, and consider the function $\mu_n : W_n \to [0, 1]$ defined by

$$\mu_n(v_0, \ldots, v_n) = \prod_{k=1}^{n} P(v_k \mid v_0, \ldots, v_{k-1})$$

for all walks $v_0, \ldots, v_n \in W_n$. We claim that for all n, μ_n is a probability measure on W_n.

First, it is clear that μ_n takes on values between 0 and 1, since each term in the product is between 0 and 1. We also need to check that $\mu_n(W_n) = 1$. This can be shown by induction. For $n = 0$, $\mu_0(W_0) = \mu_0(v_0) = P(v_0 \mid v_0) = 1$. Now suppose that $\mu_n(W_n) = 1$.

$W_{n+1} = \bigcup_{A \in W_n} B_A^{n+1}$, where B_A^{n+1} is the cylinder set consisting of all walks of length $n+1$ that start with A. Therefore

$$\begin{aligned}
\mu_{n+1}(W_{n+1}) &= \mu_{n+1}(\bigcup_{A \in W_n} B_A^{n+1}) = \sum_{A \in W_n} \mu_{n+1}(B_A^{n+1}) = \\
&= \sum_{A \in W_n} \sum_u P(u \mid A) \mu_n(A) = \sum_{A \in W_n} \mu_n(A) = \\
&= \mu_n(\bigcup_{A \in W_n} A) = \mu_n(W_n) = 1
\end{aligned}$$

Finally, we must also check that these measures satisfy the compatibility condition,

$$\mu_{n-1}(v_0, \ldots, v_{n-1}) = \sum_u \mu_n(v_0, \ldots, v_{n-1}, u).$$

We have

$$\begin{aligned}
\mu_{n-1}(v_0, \ldots, v_{n-1}) &= \prod_{k=1}^{n-1} P(v_k \mid v_0, \ldots, v_{k-1}) = \\
&= \sum_u P(u \mid v_0, \ldots, v_{n-1}) \prod_{k=1}^{n-1} P(v_k \mid v_0, \ldots, v_{k-1}) = \\
&= \sum_u \mu_n(v_0, \ldots, v_{n-1}, u)
\end{aligned}$$

Now suppose that we are given a compatible sequence $\{\mu_n\}_n$ of probability measures on $\{W_n\}_n$. Define

$$P(v_n \mid v_0, \ldots, v_{n-1}) = \frac{\mu_n(v_0, \ldots, v_n)}{\mu_{n-1}(v_0, \ldots, v_{n-1})}.$$

Then

$$\sum_u P(u \mid v_0, \ldots, v_n) = \frac{\sum_u \mu_{n+1}(v_0, \ldots, v_n, u)}{\mu_n(v_0, \ldots, v_n)} = \frac{\mu_n(v_0, \ldots, v_n)}{\mu_n(v_0, \ldots, v_n)} = 1$$

by the compatibility condition on the measures μ_n. □

Note that the values $P(v_{n+1} \mid v_0, v_1, \ldots, v_n)$ of patrolling scheme P agree with the conditional probabilities $\mathbb{P}(v_{n+1} \mid v_0, v_1, \ldots, v_n) = \dfrac{\mu_{n+1}(v_0, v_1, \ldots, v_{n+1})}{\mu_n(v_0, v_1, \ldots, v_n)}$.

Definition 7.5. To any patrolling scheme P we associate a probability measure μ on the space W_∞ of infinite walks beginning at v_0 by defining

$$\mu\left(\bigcup_{i=1}^n B(\sigma_i)\right) = \sum_{i=1}^n \mu_{i_k}(\sigma_i),$$

where i_k is the length of the walk σ_i, and μ_{i_k} is defined as in Definition 7.3. We call such a probability measure the **associated measure** for P.

Theorem 7.4. *For all probability measures μ on W_∞, there exists a unique patrolling scheme P for which μ is the associated measure.*

Proof. First suppose that we are given a patrolling scheme P. Then we can define the associated sequence $\{\mu_n\}_n$ of probability measures on the spaces $\{W_n\}_n$. We define an algebra $\mathcal{A}$ consisting of the finite unions of cylinder sets, and the complements thereof. Define a function λ in the following way. Let $\{B(\sigma_i)\}$ be a disjoint set of cylinder sets. Then $\lambda\left(\bigcup_{i=1}^n B(\sigma_i)\right) = \sum_{i=1}^n \mu_{i_k}(\sigma_i)$, where i_k is the length of the walk σ_i. By Theorem 1.14 in [18], λ extends uniquely to a measure μ on the σ-algebra generated by $\mathcal{A}$.

Conversely, suppose that we are given a probability measure μ on W_∞. Then define a function P by

$$P(v_n \mid v_0, \ldots, v_{n-1}) = \frac{\mu(B(v_0, \ldots, v_n))}{\mu(v_0, \ldots, v_{n-1})}.$$

We have that

$$\begin{aligned}
\sum_u P(u \mid v_0, \ldots, v_n) &= \sum_u \frac{\mu(B(v_0, \ldots, v_n, u))}{\mu(B(v_0, \ldots, v_n))} = \\
&= \frac{1}{\mu(B(v_0, \ldots, v_n))} \sum_u \mu(B(v_0, \ldots, v_n, u)) = \\
&= \frac{1}{\mu(B(v_0, \ldots, v_n))} \mu\left(\bigcup_u B(v_0, \ldots, v_n, u)\right) = \\
&= \frac{1}{\mu(B(v_0, \ldots, v_n))} \mu(B(v_0, \ldots, v_n)) = 1
\end{aligned}$$

□

7.3.4 Optimal patrolling schemes

In this context, we assume that G is a connected, finite graph of size at least 3. We would like to define a continuous real-valued function Q^* on $P(G)$, the space of all patrolling schemes on a graph G, that evaluates how good a patrolling scheme is. Since $P(G)$ is equivalent to the space $\mathcal{P}$ of probability measures on W_∞, and we know that W_∞ is a compact metric space, $P(G)$ is compact under the weak-* topology. Therefore Q^* must attain a maximum on $P(G)$, which says that there must exist an optimal patrolling scheme.

To make this mathematically rigorous, we first need several definitions. Let $P \in$

$P(G)$ be a patrolling scheme, μ the associated measure for P, and $H \in \overline{W}$ the finite walk that the cop has taken from time 0 until now (time t) – i.e. the cop's **history**. Then

- Define $d(H, v) = \sum_u \frac{\mu_{t+u}(\text{ Cop hits } v \text{ at time } t+u \mid H)}{u}$. This is the danger for the robber of starting a crime (now) at vertex v.
- Define $D(H) = \min_v \{d(H, v)\}$. This measures the danger for the robber given that he picks the best vertex at which to start his crime.
- Define $Q(P) = \inf_{H:\mu_t(H)>0} \{D(H)\}$. This measures the "quality" of P, in the sense that the lower $Q(S)$, the lower the least danger is for the robber over all possible histories of the cop.

This measurement of the quality of a patrolling scheme is somewhat intuitive: we are looking at how low the danger can be for the robber over all possible paths that the cop might take under her chosen patrolling scheme. However, we need our quality function to be continuous in order to be able to use the compactness of $P(G)$ to prove that it attains a maximum. Consider a patrolling scheme R' that is very similar to, say, the simple random walk, R. The only difference between R and R' is that in the case of a very low-probability event, there is a vertex that the cop with patrolling scheme R' never revisits. Then the we would expect $Q(R') = 0$ and $Q(R) > 0$, though R and R' are very close together. Therefore, it seems that Q fails to be continuous.

Before we formalize this argument, we must formalize the notion we are using of the distance between two patrolling schemes. Instead of looking at the patrolling schemes directly, we will consider a metric on the space of probability measures, and then refer to the equivalence theorem when it is more convenient to talk about a patrolling scheme.

Definition 7.6. Let μ and μ' be two probability measures on W_∞. Then define the distance between them $d(\mu, \mu')$ to be

$$d(\mu, \mu') = \max_E \mid \mu(E) - \mu'(E) \mid$$

where the maximum is taken over all events E.

Claim 7.1. *The function Q is not continuous.*

Proof. Let X_n be the union of all cylinder sets generated by walks of length $2n$ for which there is a vertex v that is visited at least n times. Let R be the simple random walk on our graph G and for each n, define the patrolling scheme R_n in the following way.

$$R_n(i \mid W) = \begin{cases} R(i \mid W) & \text{if } W \notin X_n \\ \\ \dfrac{R(i \mid W)}{1 - R(v \mid W)} & \text{if } W \in X_n \text{ and } i \neq v \\ \\ 0 & \text{if } W \in X_n \text{ and } i = v \end{cases}$$

where $R_n(i \mid W)$ corresponds to the conditional probability of coming to state i (in one step) given that the cop has traveled the walk W.

We must first check that R_n is a patrolling scheme.

If $W \in X_n$,

$$\sum_i R_n(i \mid W) = \sum_i \frac{R(i \mid W)}{1 - R(v \mid W)} =$$

$$= \sum_{i \neq v} \frac{R(i \mid W)}{1 - R(v \mid W)} =$$
$$= \frac{1 - R(v \mid W)}{1 - R(v \mid W)} = 1$$

where i ranges over all states i reachable in one step after the walk W, and v is a state that was hit at least n times in the first $2n$ steps of W. If $W \notin X_n$ then

$$\sum_i R_n(i \mid W) = \sum_i R(i \mid W) = 1.$$

Now let σ, σ_n be the measures associated to R, R_n, respectively. Then

$$\begin{aligned} d(\sigma_n, \sigma) &= \max_E \mid \sigma_n(E) - \sigma(E) \mid = \\ &= \max_E(\sigma_n(E) - \sigma(E)) = \\ &= \max_E(\sigma_n(E \cup X_n) + \sigma_n(E \backslash X_n) - \sigma(E \cup X_n) - \sigma(E \backslash X_n)) = \\ &= \max_E(\sigma_n(E \cup X_n) - \sigma(E \cup X_n)) \leq \\ &\leq \max_E(\sigma_n(X_n) - 0) < \frac{1}{2^n}. \end{aligned}$$

The last inequality follows from the fact that the highest probability for the traversal of each edge is 1/2, except in the case of the path consisting of three vertices and two edges, in which case the vertex in the middle of the path has a probability of $\frac{1}{2^n}$ of being visited at least n times in the first $2n$ steps since every other edge will be traversed with probability 1. Therefore we have that $\lim_{n \to \infty} d(\sigma_n, \sigma) = 0$, and so the sequence $\{\sigma_n\}$ has the probability measure σ as its limit.

Now what remains to be shown is that $Q(\sigma) > 0$ and $Q(\sigma_n) = 0$. For the latter, we need to see that X_n occurs with positive probability under the patrolling scheme R_n. This is equivalent to showing this happens under R. If the cop starts at vertex v_0, then she goes to a particular vertex, say u, with probability $\frac{1}{deg(v_0)}$. Similarly, she goes back to v_0 with probability $\frac{1}{deg(u)}$. Therefore, X_n occurs with probability at least $\frac{1}{deg(v_0)^n}\frac{1}{deg(u)^n} > 0$. Therefore, if X_n occurs and v is the vertex from the description of the event X_n,

$$Q(\sigma_n) = \inf_{H:\mathbb{P}(H)>0}\{D(H)\} \leq \sum_u \frac{\mu_{2n+u}(\text{ Cop hits } v \text{ at time } 2n+u \mid X_n)}{u} = 0$$

However,

$$\begin{aligned} Q(\sigma) &= \inf_{H:\mathbb{P}(H)>0}\{D(H)\} = \\ &= \inf_{x:\mathbb{P}(v_t=x)>0} \min_v \sum_u \frac{\mu_u(\text{ Cop hits } v \text{ at time } t+u \mid \text{Cop starts at } x)}{u} \\ &> \min_{x:\mathbb{P}(v_t=x)>0} \min_v \frac{\mu_L(\text{a shortest walk from } x \text{ to } v)}{L} > 0 \end{aligned}$$

where L is the length of a shortest walk from x to v_t.

Therefore we have a sequence of probability measures σ_n, each with $Q(\sigma_n) = 0$, the converge to a measure σ with $Q(\sigma) > 0$, and so Q fails to be continuous. □

Therefore we must make some adjustments in order to turn this notion of quality into a continuous one. There are a number of ways by which we could take care of this issue: for instance, only considering patrolling schemes with bounded cover time (i.e. schemes in which there exists a bound, M, on how many moves it can take the cop to visit all of the vertices in the graph) or only considering patrolling schemes with bounded memory. We will consider the latter here, and there is reason to hope that this restriction does not actually lose us anything.

Conjecture 7.1. *On any graph G, an optimal scheme has bounded memory.*

7.3.5 Example: Optimal patrolling scheme on K_3

Definition 7.7. For a sequence S of vertices in K_3 let α_S be the probability of continuing in the same directions as the last step of S, given that the cop has just walked S. Let $D(v \mid S)$ be the danger (for the robber) of committing a crime now at vertex v given that the cop has just walked S.

Note that if we define S_i to be the walk satisfied by the recursion

$$S_2 = ab, S_{2i+1} = S_{2i}c, S_{2i+2} = S_{2i+1}b$$

and let $\alpha_i := \alpha_{S_i}$ then by definition, we have the following:

$$\begin{aligned} D(a \mid ab) &= \frac{1-\alpha_2}{1} + \frac{\alpha_2\alpha_3}{2} + \frac{\alpha_2(1-\alpha_3)\alpha_4}{3} + \frac{\alpha_2(1-\alpha_3)(1-\alpha_4)\alpha_5}{4} + \dots \\ &= 1 - \alpha_2 + \sum_{k \geq 2} \frac{\alpha_2\alpha_{k+1}}{k} \prod_{j=3}^{k} 1 - \alpha_j \end{aligned}$$

Now define D_r to be $D(a \mid ab)$ with all α_i replaced by r for some $r \in [\frac{1}{2}, 1]$, so $D_r = 1 - r + \left(\frac{r}{1-r}\right)^2 \left(\ln\left(\frac{1}{r}\right) - 1 + r\right)$. Then we have the following fact:

Lemma 7.1. *If $\alpha_2 = r$ for some $r \in [\frac{1}{2}, 1]$ and for all $i > 2$ we have that $\alpha_i \in [1-r, r]$ then $D(a \mid ab) \leq D_r$.*

Proof. Note first that $D_r = 1 - r + \sum_{k\geq 2} \frac{r^2(1-r)^{k-2}}{k}$. The numerators in the terms of $D(a \mid ab)$ sum to 1, that is

$$1 - \alpha_2 + \alpha_2\alpha_3 + \alpha_2(1-\alpha_3)\alpha_4 + \cdots = 1.$$

$\alpha_3 \leq \alpha_2 = r$, so if increasing α_i to r increases $\alpha_2\alpha_3$ by a total of q, then it must decrease the sum of the numerators of the other terms by the same amount; that is, the sum

$$\alpha_2(1-\alpha_3)\alpha_4 + \alpha_2(1-\alpha_3)(1-\alpha_4)\alpha_5 + \ldots$$

decreases by q. But then the total change is greater than $\frac{q}{2} - \frac{q}{k}$ for some $k > 3$, making the total change positive.

Therefore, we have that maximizing the α_i in the interval $[1-r, r]$ maximizes the value of $D(a \mid ab)$. □

Using this lemma, we have the following result about patrolling schemes on K_3, the complete graph on 3 vertices.

Theorem 7.5. *The random walk is an optimal patrolling scheme on K_3.*

That is, it does not pay for the cop to increase the length of her memory while patrolling K_3.

Proof. The random walk has quality $\displaystyle\sum_{k=1}^{\infty} \frac{1}{k2^k} = \ln(2)$. (Note that this equals D_r with $r = \frac{1}{2}$.)

D_r has first derivative $\dfrac{2r\ln(r) - r^2 + 1}{r^3 - 3r^2 + 3r - 1}$, which is negative for all $r \in [\frac{1}{2}, 1]$. Therefore D_r is in fact maximized at $r = \frac{1}{2}$, so the danger for the robber of committing the crime behind the cop, $D(a \mid ab) \leq \ln(2)$. A similar arguments shows that

$D(c \mid ab) \leq \ln(2)$, as well. Thus the quality of any patrolling scheme on K_3 cannot exceed that of the random walk. □

Note that equality holds in the inequality in Lemma 7.1 if and only if $\alpha_3 = 0$, in which case the difference $q = |\alpha_2\alpha_3 - r^2| = 0$.

Bibliography

[1] M. Adler, H. Räcke, N. Sivadasan, C. Sohler, and B. Vöcking, *Randomized pursuit-evasion in graphs*, Comb., Prob., and Comp. **12** (2003), no. 3, 225–244.

[2] N. Alon, P. Pralat, and N. C. Wormald, *Cleaning regular graphs with brushes*, SIAM Journal on Discrete Mathematics **23** (2008), 223–250.

[3] B. Alspach, *Searching and sweeping graphs: A brief survey*, Le Matematiche **54** (2004), 5–37.

[4] B. Alspach, D. Dyer, D. Hanson, and B. Yang, *Time constrained graph searching*, Theoretical Computer Science **399** (2008), 158–168.

[5] T. Andreae, *On a pursuit game played on graphs for which a minor is excluded*, J. Combin. Theory Ser. B. **41** (1986), no. 1, 37–47.

[6] Y. Babichenko, Y. Peres, R. Peretz, P. Sousi, and P. Winkler, *Hunter, Cauchy rabbit, and optimal Kakeya sets*, Trans. Amer. Math. Soc., to appear, 2013.

[7] T. Basar, G. J. Olsder, and E. M. Reingold, *"Lion and Man": upper and lower bounds*, ORSA J. Comput. **4** (1992), 447–452.

[8] A. Berarducci and B. Intrigila, *On the cop number of a graph*, Adv. Appl. Math. **14** (1993), 389–403.

[9] A. Beveridge, Private communication, 2nd Annual GRASCan Workshop, Ryerson University, April 2013.

[10] A. Bonato, E. Chiniforooshan, and P. Pralat, *Cops and robbers from a distance*, **411** (2010), no. 43, 3834–3844.

[11] A. Bonato, P. Golovach, G. Hahn, and J. Kratochvil, *The capture time of a graph*, Discrete Math. **309** (2009), 5588–5595.

[12] G.R. Brightwell and P. Winkler, *Maximum hitting time for random walks on graphs*, J. Rand. Struct. and Alg. **1** (1990), no. 3, 263–276.

[13] ______, *Gibbs measures and dismantlable graphs*, J. Comb. Theory (Series B) **78** (2000), 141–169.

[14] ______, *Hard constraints and the bethe lattice: adventures at the interface of combinatorics and statistical physics*, Proc. Int'l. Congress of Mathematicians Vol. III, Li Tatsien, ed. (2002), 605–624.

[15] W. H. Bussey and O. Veblen, *Finite projective geometries*, Trans. Amer. Math. Soc. **7** (1906), 241–259.

[16] D. Coppersmith, P. Tetali, and P. Winkler, *Collisions among random walks on a graph*, SIAM J. Disc. Math. **6** (1993), no. 3, 363–374.

[17] S. L. Fitzpatrick and R. J. Nowakowski, *Copnumber of graphs with strong isometric dimension two*, Ars Combin. **59** (2001), 65–73.

[18] G. B. Folland, *Real analysis: Modern techniques and their applications*, John Wiley & Sons, Inc, USA, 1999.

[19] F. Fomin and P. Golovach, *Graph searching and interval completion*, SIAM Journal on Discrete Mathematics **13** (2000), 454–464.

[20] F. Fomin, P. Golovach, J. Kratochvil, N. Nisse, and K. Suchan, *Pursuing a fast robber on a graph*, Theoretical Computer Science **411** (2010), no. 7-9, 1167–1181.

[21] P. Frankl, *Cops and robbers in graphs with large girth and cayley graphs*, Disc. Appl. Math. **17** (1987), no. 3, 301–305.

[22] A. Frieze, M. Krivelevich, and P. Loh, *Variations on cops and robbers*, Journal of Graph Theory **69** (2012), 383–402.

[23] Shmuel Gal, *Search games with mobile and immobile hider*, SIAM J. Control & Opt. **17** (1979), no. 1, 99–122.

[24] T. Gavenčiak, *Games on graphs*, Master's thesis, Charles University, Prague, 2007.

[25] G. Hahn, F. Laviolette, N. Sauer, and R.E. Woodrow, *On cop-win graphs*, Discrete Math. **258** (2002), 27–41.

[26] ______, *On cop-win graphs*, Discrete Math. **258** (2002), 27–41.

[27] G. Hahn and G. MacGillivray, *A note on k-cop, l robber games on graphs*, Discrete Math. **306** (2006), 2492–2497.

[28] M. Hall, *Projective planes*, Trans. Amer. Math. Soc. **54** (1943), 229–277.

[29] D. Hess, Private communication, 2012.

[30] R. Isaacs, *Differential games: A mathematical theory with applications to warfare and pursuit, control and optimization*, John Wiley & Sons, Inc, New York, USA, 1965.

[31] M. S. Klamkin, *A slow ship intercepting a fast ship*, Amer. Math. Monthly **59** (1952), no. E991, 408.

[32] R. C. Larson, *Urban police patrol analysis*, MIT Press, USA, 1972.

[33] L. Lu and X. Peng, *On Meyniel's conjecture of the cop number*, J. Graph. Th. **71** (2012), 192–205.

[34] G. MacGillivray, Private communication, May 2011.

[35] S. McKeil, *Chip firing cleaning processes*, 2007.

[36] D. Merlini, R. Sprugnoli, and M.C. Verri, *An analysis of a simple algorithm for random derangements,*, Proceedings of the 10th Italian Conference on Theoretical Computer Science (2007), 139–150.

[37] M. Messinger, R. J. Nowakowski, and P. Pralat, *Cleaning with brooms*, Graphs and Combinatorics **27** (2011), 251–267.

[38] M. Messinger, R. J. Nowakowski, P. Pralat, and N. Wormald, *Cleaning random d-regular graphs with brushes using a degree-greedy algorithm*, Proceedings of the 4th Workshop on Combinatorial and Algorithmic Aspects of Networking (2007).

[39] M.E. Messinger, R. Nowakowski, and P. Pralat, Theoretical Computer Science **399** (2008), 191–205.

[40] N. Nisse and K. Suchan, *Fast robber in planar graphs*, Graph Theoretic Concepts in Computer Science: 34th International Workshop (2008).

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

[42] T.D. Parsons, *Pursuit-evasion in a graph*, Theory and Applications of Graphs (1976), 426–441.

[43] R. Peyre, *A probabilistic approach to Carne's bound*, Potential Anal. **29** (2008), no. 1, 17–36.

[44] A. Quilliot, *Homomorphismes, points fixes, rétractations et jeux de poursuite dans les graphes, les ensembles ordonnés et les espaces métriques*, Ph.D. thesis, Université de Paris VI., 1983.

[45] J. Soares, *Maximum diameter of regular digraphs*, J. Graph Theory **16** (1992), 437–450.

[46] J. Urrutia, *Handbook of computational geometry*, Elsevier, Amsterdam, 2000.

[47] N. Th. Varopoulos, *Long range estimations for markov chains*, Bull. Sci. Math **109** (1985), 225–252.

[48] A. Wald, *On cumulative sums of random variables*, The Annals of Mathematical Statistics **15** (1944), no. 3, 283–296.