THE EFFECTS OF REPLACING COPS AND SEARCHERS WITH TECHNOLOGY

By

Nancy Clarke

SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
AT
DALHOUSIE UNIVERSITY
HALIFAX, NOVA SCOTIA
AUGUST 1999

0-612-49330-X

For my nan

with all my love

and admiration.

Contents

List of Tables

List of Figures

Abstract

In this thesis, we examine some well known pursuit and evasion games. The focus will be on the game of Cops and Robber, introduced by Nowakowski and Winkler [10], and independently, by Quilliot [12], and the searching game, introduced by Parsons [11].

We proceed to introduce some variations of these games. We alter the amount of information available to the cops or searchers regarding the position of the robber. We also replace some of the dynamic cops or searchers by static objects. As well, we consider the searching game from the perspective of minimizing the time expected for the search.

Acknowledgements

I would like to thank my supervisor. Dr. Richard Nowakowski for his guidance. and for sharing not only his time and knowledge but also his enthusiasm. I would also like to thank the NSERC for its financial support.

I would like to thank my supervisor during my time at the NRC. Dr. Martin Brooks for offering sound advice and considerable encouragement. I would also like to thank the math department at Sir Wilfred Grenfell College for helping me to discover mathematics. and for opening so many doors.

I would like to thank my parents. George and Christine. my greatest supporters and best friends: my grandmother. Mrs. Eliza Blanche Francis for always believing in me. and for teaching me things that cannot be learned in university: my brother. Rodney for challenging me as we were growing up: and my brother. Adam for reminding me that there are things in life more important than my thesis.

Chapter 1

Introduction

In this thesis, some well known searching games are examined: in particular the game of Cops and Robber is considered. These games have traditionally been dynamic. In Cops and Robber, the opponents are able to move from vertex to vertex as the cops attempt to apprehend the robber. A game known as searching is usually formulated in terms of clearing an area of certain airborne contaminants, or alternately, searching for an infinitely fast intruder. We will think of this game as a variation of Cops and Robber in which the cops take on the role of the searchers. Both sides move continuously. The Watchman Problem requires that the vertices of the underlying graph G be monitored rather than searched: that is, for every vertex v of G, v or one of its neighbors must be visited. In addition, a time period is introduced. The vertices of G must be monitored once every k units of time, $k > 0$.

In this first chapter, we look at the historical development of these games, and present some known results. In the process of giving results regarding the outcomes of these games, we look at some strategies that can be used by a particular side to guarantee a favorable outcome. In so doing, we notice that the dynamic nature of the players is not always utilized in winning strategies. This realization led us to alter the rules of the games, and investigate the use of static objects in place of the dynamic cops. This is especially evident in the third chapter where we propose replacing the cops by a single cop with a finite number of 'traps' at his disposal. If the robber moves onto a vertex with a trap, he is detained and a win for the cops results.

A second theme that runs through this thesis is the amount of information available to the opponents about each other. In the game of Cops and Robber, both sides have perfect information. Each side is always aware of the position of the other. The searching game, however, is on the opposite end of the spectrum. The cops have no information regarding the position of the robber; in fact, the cops are not even certain that there is a robber. We propose some variations of these games that fall somewhere in between. The second chapter considers the use of special surveillance equipment. The only information the cops have about the robber is that which can be obtained using this equipment. We present two variations, the first being the use of photo radar. If the robber moves using an edge equipped with photo radar, the cops will be aware of the movement regardless of their positions. The second is the use of surveillance cameras on a grid. This surveillance makes the cops aware of the position of the robber if he is located on a grid line covered by a camera. Otherwise, the cops have no information regarding the robber.

In the final chapter, the problems are formulated in terms of searchers, and lost dogs and sleeping babies. Dogs and babies are used to indicate the amount of maneuverability of the 'searchee'. We could have chosen to present these problems in terms of cops and robbers in keeping with the other chapters. However in these games, unlike those considered up to this stage, the player being searched for is unaware of the actions of the searchers. Similarly, the searchers do not know the position of this player and are unable to influence his movements, although the searchers are able to deduce the probability that the lost dog or baby is located on a particular vertex at any given time. This information is used to determine the vertex on which the search should begin in order to minimize the expected time required for the search. A variation of the lost dog problem is then introduced. The searchers are able to use static traps. Once the dog moves onto a vertex with a trap, it is detained, the searchers are aware of its position and the game is over. Probabilities are calculated and expected times are minimized for this version of the game as well.

These probability problems can also be modeled in terms of a search and rescue at sea. The searchers are in a boat rather than on foot. and are searching for a missing person. Instead of traps. the searchers have a number of lifeboats at their disposal. The searchers are able to monitor the lifeboats so that if the missing person is able to board such a boat. the search is complete.

1.1 Cops and Robber

1.1.1 Rules of the Game

The game of Cops and Robber is a pursuit game played on a graph G. This game was introduced by Nowakowski and Winkler [10] and independently. by Quilliot [12]. The game is played by two opposing sides: the cop side is composed of a set of $k > 0$ cops and the robber side is composed of a single robber. Both sides play with perfect information: that is. each side knows the whereabouts of the other at all times. The rules require that the cops begin the game by each choosing a vertex to occupy. These vertices do not have to be distinct. The robber must then also choose a vertex to occupy. The opponents move alternately. It should be noted that a player can choose to pass and remain where he is during a turn. Hence during the cops' turn. it is allowable for only a subset of the k cops to move. The cops win if at least one of them occupies the same vertex as the robber after a finite number of moves. The robber wins if this situation can be avoided forever. We note that in this game. unlike searching. the players are always assumed to be located on vertices.

The version of the game described here allows both sides to pass during a turn if they so desire. This is known as the **passive** game. In the **active** game. the robber and a nonempty subset of the cops must move during their respective turns. It has been shown by Neufeld [8] that if k cops have a winning strategy on a graph G in the passive game. then the number of cops. k' needed in the active game must satisfy $k' \leq k \leq k' + 1$. If the game is played on a **reflexive** graph. a graph with a loop at every vertex. the passive and active versions are equivalent.

1.1.2 Characterization of Copwin Graphs

When the game was originally proposed, it was played with a single cop and a robber. Any graph could be characterized as either **copwin** or robber-win depending on the outcome of the game. Copwin graphs were completely characterized in [10] and [12].

Example: Each member of the set $\{T_i : T_i$ is a finite tree$\}$ is copwin. To see this, consider the vertex occupied by the cop at any stage. The robber is unable to move past the cop because there is just a single path joining any two vertices. Hence the tree is partitioned into two parts by the cop, and the robber is restricted to moving within one of those parts. As the cop moves toward the robber, that part of the graph that is inaccessible to the robber strictly increases as the robber's portion becomes smaller. Hence after a finite number of moves, the robber is apprehended.

Example: Let C be the family of cycles of length greater than three. Each member of this family is robber-win since the robber can always stay at least two vertices away from the cop.

Definition 1.1 *Let G and H be reflexive graphs. A mapping $f : V(G) \to V(H)$ is said to be **edge preserving** if it preserves adjacencies: that is, if there is an edge joining two vertices in G, there must be an edge (possibly a loop) joining the images of these vertices in H.*

Definition 1.2 *Let G be a reflexive graph and let H be an induced subgraph of G. It is said that H is a **retract** of G if there is an edge preserving map f from G to H such that the restriction of f to H is the identity map on H.*

Theorem 1.1 (Nowakowski and Winkler [10]) *Any retract H of a copwin graph G is also a copwin graph.*

Proof. Let G be a copwin graph and let H be a retract of G. Further let f be a retraction map from G to H. Since G is copwin, the cop has a winning strategy on G. This strategy can be modified and used on the subgraph H. The cop simply plays the image under f of his winning strategy on G. Using this strategy, the cop captures the image of the robber on H. Since the robber is actually playing on H and f is the identity map on H, the robber's image coincides with his actual position. Hence the robber is apprehended on H and therefore, H is a copwin graph. □

Definition 1.3 *Let G be a graph and let $v \in V(G)$. The* ***neighborhood*** *of v, denoted $N(v)$, is the set of vertices adjacent to v in G. The* **closed** *neighborhood of v, denoted $N[v]$ is defined as $N(v) \cup \{v\}$.*

Definition 1.4 *A vertex d of a graph G is said to* ***dominate*** *another vertex v if d is adjacent to each of the vertices in the closed neighborhood of v.*

Suppose a given graph G is copwin. To determine the properties that characterize such a graph, it is useful to consider the last move made by the robber before he is apprehended. Let the position of the robber before this last move be denoted v. There are three options open to the robber. He can choose to pass and remain on vertex v, he can move onto the vertex occupied by the cop, or he can move to a vertex adjacent to the cop's position. Since all of these options must lead to the capture of the robber, it must be true that the vertex u occupied by the cop is adjacent to v and also to every vertex that is adjacent to v; that is, u dominates v. The vertex v will be referred to as a **corner** since the robber has no means of escape once he is forced to move onto this vertex.

Clearly a graph without a corner cannot be copwin. Suppose a graph G has a corner. The robber will only move onto the corner if he is forced to do so. Hence the question becomes whether or not the cop can force the robber onto the corner. This can be determined by removing the corner and determining if the resulting graph is copwin. Intuitively, the successive removal of corners from a copwin graph will result in a single vertex. This is the idea used by Nowakowski and Winkler [10] to characterize copwin graphs.

Theorem 1.2 (Nowakowski and Winkler [10]) *Let G be a graph and let c be a corner of G. Let $G' = G \setminus \{c\}$. Then G is copwin if and only if G' is copwin.*

Proof. Let G be a graph and let c be a corner of G. Let $G' = G \setminus \{c\}$. Further let d be a vertex that dominates the corner c. We wish to show that G is copwin if and only if G' is copwin.

Suppose G is copwin. Now G' is a retract of G with a retraction map f defined as follows: $f(c) = d$ and $\forall v \in V(G')$, $f(v) = v$. Thus by Theorem 1.1, G' is copwin.

Conversely suppose G' is copwin. Let f be the retraction map from G to G'. Since G' is copwin, the cop has a winning strategy on G'. If the game is actually being played on G, the winning strategy on G' can be thought of as catching the image of the robber. Now suppose this image is caught on vertex u. If $u \neq d$, then the robber's image on G' corresponds to his actual position on G. This is because f is the identity map on G'. Hence the robber is apprehended. Otherwise, the robber's image is apprehended on vertex d. Since it is known that $f(c) = f(d) = d$, the robber is on vertex c or vertex d in the graph G. If he is on d, his actual position corresponds to his image and he is caught. If he is on c then he will be caught on the next move. This is because the cop is on vertex d and it is known that d dominates c. □

At this stage some needed definitions will be introduced.

Definition 1.5 *Let G be a graph and let $v \in V(G)$. Suppose there exists a vertex $u \in V(G)$ such that $N[v] \subset N[u]$. Then v is said to be* ***irreducible****. The vertex v is also known as a corner or pitfall.*

Definition 1.6 *A graph G is said to be* ***dismantlable*** *if there is an ordering $\{v_1, v_2, \ldots, v_n\}$ of the vertices of G such that for each $i < n$, v_i is irreducible in the subgraph induced by $\{v_i, v_{i+1}, \ldots, v_n\}$.*

We are now ready to give the the main theorem in this section.

Theorem 1.3 (Nowakowski and Winkler [10]) *A finite graph is copwin if and only if it is dismantlable.*

The ordering of the vertices of the graph G referred to in the previous definition is known as a **copwin ordering**.

Example: This example refers to Figure 1.1. The circled vertices represent corners at each of the stages. Also, at each stage, it does not matter in which order the corners are removed. The original graph is copwin by Theorem 1.3.

There are copwin graphs that are not finite. So Nowakowski and Winkler [10] extended Theorem 1.3 to obtain a complete characterization of copwin graphs. However, we will not be considering infinite graphs and therefore omit this proof.

1.1.3 A Variety

Now that copwin graphs have been characterized, a property of such graphs will be explored. We begin with some definitions.

Definition 1.7 *The **strong product** of a set of graphs $\{G_i : i = 1, 2, \dots, k\}$ is the graph $\boxtimes_{i=1}^{k} G_i$ whose vertex set is the Cartesian product of the sets $\{V(G_i) : i = 1, 2, \dots, k\}$, and there is an edge between $\overline{a} = (a_1, a_2, \dots, a_k)$ and $\overline{b} = (b_1, b_2, \dots, b_k)$ if and only if a_i is adjacent or equal to b_i for all $i = 1, 2, \dots, k$.*

Definition 1.8 *A **variety** of graphs is a class of graphs which is closed under finite products and retracts.*

The next theorem has been proven by Nowakowski and Winkler [10].

Theorem 1.4 (Nowakowski and Winkler [10]) *Let $\{G_i : i = 1, 2, \dots, k\}$ be a finite collection of copwin graphs. The strong product of these graphs is also copwin.*

Proof. Let $\{G_i : i = 1, 2, \dots, k\}$ be a finite collection of copwin graphs. Let $G = \boxtimes_{i=1}^{k} G_i$ be the strong product of these graphs. We wish to show that G is copwin.

There is an edge-preserving projection of G onto each of the graphs G_i. Hence the cop and robber can be projected onto each of the original graphs and a game can take place there. Consider one such projection onto the graph G_i. Now G_i is copwin, and so the cop has a winning strategy and is able to apprehend the robber. In terms of the

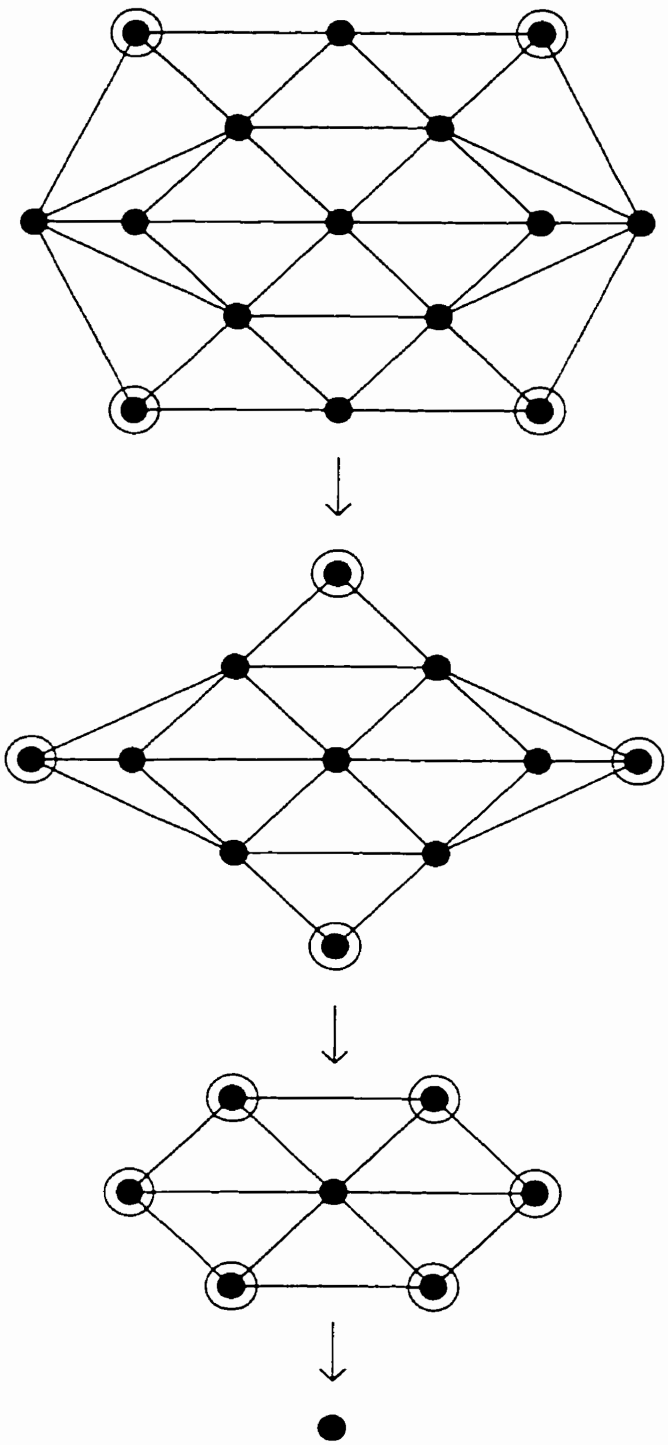

Figure 1.1: An illustration of dismantling. The original graph is copwin.

larger graph G, the cop has apprehended the projection of the robber on G_i. The cop stays with this projection for the remainder of the game as he similarly captures the other projections of the robber on the graphs $G_1, \ldots, G_{i-1}, G_{i+1}, \ldots, G_k$. Since the collection of graphs is finite, the robber will be apprehended on each of the projections after a finite number of moves. At this time, the robber is apprehended on G. It should be noted that the cop has played the composition of his winning strategies on each of the graphs G_i. □

Example: The previous theorem tells us that the product of a finite collection of copwin graphs is copwin. This example illustrates why the theorem cannot be extended to infinite collections of copwin graphs.

Define $P_n = \{0, 1, 2, \ldots, n-1\}$. Now P_n is copwin. Consider the product of an infinite collection of such paths $\boxtimes_{i=1}^{\infty} P_i$. This graph is not copwin since the vertices $(0, 0, 0, \ldots)$ and $(0, 1, 2, \ldots)$ are not connected by a finite path.

The next theorem can be found in papers by Aigner and Fromme [1], and by Nowakowski and Winkler [10]. It follows immediately from Theorem 1.1 and Theorem 1.4.

Theorem 1.5 (Aigner and Fromme [1] and Nowakowski and Winkler [10]) *The class of copwin graphs is a variety.*

1.1.4 Bridged Graphs

Bridged graphs and their relationship to copwin graphs will now be considered.

Definition 1.9 *Let G be a graph and let H be a subgraph of G. The graph H is said to be* ***isometric*** *if the distance between any pair of vertices in H is the same as that in G.*

Clearly an isometric subgraph of a graph G must be an induced subgraph.

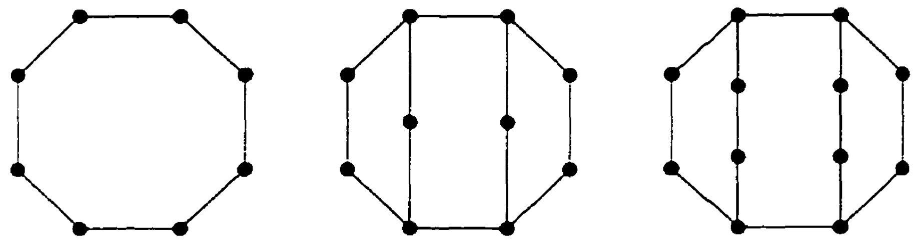

Figure 1.2: The graph H is an isometric subgraph of G but not of I.

Definition 1.10 *Let G be a graph and let $x, y \in V(G)$. Let C be a cycle of length at least four in G that contains the vertices x and y. If $d_C(x, y) > d_G(x, y)$ then the graph G is said to be* ***bridged****.*

Example: See Figure 1.2 for examples of these graphs. The graph H is an isometric subgraph of G but not of I. The graph I is bridged.

In effect, this definition says that if C is any cycle with length greater than three in a bridged graph G, then there is a 'shortcut' between a pair of vertices on the cycle.

Anstee and Farber [2] published a paper concerning bridged graphs and copwin graphs. The paper begins by proving a result that tells us every nontrivial bridged graph contains a corner. The paper goes on to prove the next theorem.

Theorem 1.6 (Anstee and Farber [2]) *Let G be a bridged graph. There exists a vertex $u \in V(G)$ such that $G \setminus u$ is bridged.*

Proof. Let G be a bridged graph. Choose a pair of vertices $u, v \in V(G)$ such that $N[u] \subset N[v]$. Such a pair is known to exist [2]. Now let P be a shortest path in G that contains u but in which u is not a leaf. Now u can be replaced by v in this path. Hence $G \setminus u$ is an isometric subgraph of G. It is noted that a cycle C is isometric in $G \setminus u$ if and only if it is isometric in G. Therefore, $G \setminus u$ is also a bridged graph. $\square$

In the proof of the theorem, the vertex u was taken to be a corner. Hence the theorem actually tells us that the removal of a corner from a bridged graph results in another bridged graph.

Let's collect the information given us through several theorems. First, it is known that every bridged graph contains a corner, and that its removal results in another bridged graph. It is also known that a copwin graph is one in which the successive removal of corners results in a single vertex. Hence we are able to conclude, as Anstee and Farber did, that every bridged graph is copwin. This result is stated as a theorem.

Theorem 1.7 (Anstee and Farber [2]) *Let G be a bridged graph. Then G is copwin.*

An algorithmic proof of this theorem has been given by Chepoi [4]. He has shown that every ordering of the vertices of a bridged graph produced by a breadth-first search is a copwin ordering as defined by Nowakowski and Winkler.

1.1.5 Strategy for a Copwin Graph

Suppose $\{x_1, x_2, \ldots, x_n\}$ is a copwin ordering of the vertices of a graph G. We know that the cop must have a winning strategy on G. But this strategy has not been made explicit. The goal of this section is to describe a strategy, which appears to be new, that can be used by the cop to win, and to prove that this strategy is effective in capturing the robber. This strategy will be very useful in Chapter 3.

Copwin Strategy. Let $\{x_1, x_2, \ldots, x_n\}$ be a copwin ordering of the vertices of a graph G. Define the induced subgraphs $G_i = G_{i-1} \setminus \{x_{i-1}\}$ where $G_1 = G$, and let $f_i : G_i \to G_{i+1}$ be the retraction map from G_i to G_{i+1}. The robber is always thought to be playing on the graph G. However, the cop initially moves on the subgraph G_n. The cop begins on vertex x_n, the vertex on which the cop's position coincides with the robber's image under the mapping $f_{n-1}f_{n-2}\ldots f_3 f_2 f_1$. Now suppose the cop is occupying the robber's image in the subgraph G_i under the mapping $f_{i-1}f_{i-2}\ldots f_3 f_2 f_1$.

The cop is able to move onto the image of the robber in G_{i-1}. After at most n moves, the robber is apprehended.

Proof. The robber begins the game on some vertex of G, and the cop begins the game in the subgraph G_n on vertex x_n. Now consider the image or shadow of the robber under the mapping $f_{n-1}f_{n-2}\dots f_3f_2f_1$. This mapping takes all vertices in G to x_n. Hence the cop's position coincides with the robber's image under this mapping.

Suppose the cop is playing in the subgraph G_i, and is occupying the robber's shadow under the mapping $f_{i-1}f_{i-2}\dots f_3f_2f_1$. We wish to show that the cop is able to move onto the robber's shadow in G_{i-1} under $f_{i-2}f_{i-3}\dots f_3f_2f_1$. Consider the mapping $f_{i-1}: G_{i-1} \to G_i$. We know that x_{i-1} is a corner in G_{i-1}. Let d be a vertex that dominates x_{i-1} where $d \in \{x_i, x_{i+1}, \dots, x_n\}$. The mapping f_{i-1} is defined as follows: $f_{i-1}(x_{i-1}) = d$ and $\forall v \in V(G_i), f_{i-1}(v) = v$.

Since the cop has captured the shadow of the robber in G_i, the cop must have a winning strategy on this subgraph. Suppose the robber's shadow is captured on vertex u, $u \in \{x_i, x_{i+1}, \dots, x_n\}$. If $u \neq d$, then the robber's shadow on G_i under $f_{i-1}f_{i-2}\dots f_3f_2f_1$ corresponds to his shadow on G_{i-1} under $f_{i-2}f_{i-3}\dots f_3f_2f_1$ since $\forall u \neq x_{i-1}, f_{i-1}(u) = u$. Hence the robber's shadow is captured on G_{i-1}. Otherwise, the robber's shadow is apprehended on vertex d. Recall $f_{i-1}(x_{i-1}) = f_{i-1}(d) = d$, and so the robber's shadow is on vertex x_{i-1} or vertex d in the graph G_{i-1}. If it is on d, his shadow on G_{i-1} corresponds to his shadow on G_i, and so his shadow on G_{i-1} has been apprehended. If his shadow is on x_{i-1} in the graph G_{i-1}, it will be captured on the next move. This is because the cop is on vertex d and d dominates x_{i-1}.

Since there are only a finite number of graphs G_i, the robber's shadow will coincide with his actual position after a finite number of moves. Hence the strategy presented here will result in a win for the cop. □

It has been shown that if the cop is playing in the subgraph G_i, and is occupying the robber's shadow under the mapping $f_{i-1}f_{i-2}\dots f_3f_2f_1$, then the cop is able to move onto the robber's shadow in G_{i-1} under $f_{i-2}f_{i-3}\dots f_3f_2f_1$. As a consequence, if the cop is playing in the subgraph G_i, the robber can never move to a vertex in this subgraph without being apprehended by the cop. Equivalently, the robber cannot move onto vertices used previously by the cop. This is stated as the next theorem.

Theorem 1.8 *Suppose the cop is playing the Copwin Strategy in the subgraph G_i, and is occupying the robber's shadow under the mapping $f_{i-1}f_{i-2}\dots f_3f_2f_1$. The robber can never move to a vertex of G_i without the cop immediately landing on the same vertex.*

Proof. Suppose the cop is playing in the subgraph G_i, and is occupying the robber's shadow under the mapping $f_{i-1}f_{i-2}\dots f_3f_2f_1$. The cop is able to move so as to stay with the shadow of the robber on this subgraph. Now the mapping $f_{i-1}f_{i-2}\dots f_3f_2f_1$ is the identity on G_i. Hence if the robber moves to a vertex of G_i, his shadow will correspond to his actual position and he will be apprehended. □

1.1.6 Cops and Robber with k Cops

It is evident that there are many graphs which are not copwin. A natural question to pose when considering such a graph G is how many cops are needed to apprehend the robber. This question leads to the following definition.

Definition 1.11 *Let G be a graph. The minimum number of cops needed to apprehend a robber on this graph is known as the* **copnumber** *of G and is denoted $c(G)$.*

Shortly after the introduction of the game by Nowakowski and Winkler, Aigner and Fromme [1] published results important to further study. It was they who first introduced the notion of copnumber, along with several interesting results.

Aigner and Fromme were able to prove that there exists an n-regular graph without 3- or 4-cycles for every natural number n. Using this result, they showed that there are graphs which require an arbitrary number of cops as stated in the next theorem.

Theorem 1.9 (Aigner and Fromme [1]) *Let G be a graph with minimum degree $\delta(G) \geq n$ which has no 3- or 4-cycles. Then $c(G) \geq n$.*

The most interesting result presented here involves planar graphs. We have already seen in Theorem 1.9 that there are graphs which require an arbitrary number of cops to apprehend a robber. The next theorem addresses an opposing question. It is desirable to identify a class of graphs for which a bound can be placed on the copnumber. It is known that a 3-cycle or a 4-cycle is contained in every planar graph whose minimum degree is greater than or equal to five. This relation led Aigner and Fromme [1] to conjecture the result stated in the next theorem which they have proven.

Theorem 1.10 (Aigner and Fromme [1]) *Let G be a planar graph. Then* $c(G) \leq 3$.

Example: This example refers to Figure 1.3. The graph G shown has minimum degree $\delta(G) = 3$. The smallest cycle contained in G is of length 5. Hence this graph satisfies the conditions of Theorem 1.9 and $c(G) \geq 3$. Since G is a planar graph. $c(G) \leq 3$ by Theorem 1.10. Hence $c(G) = 3$.

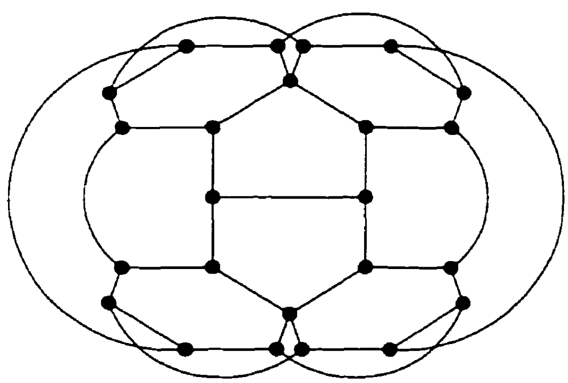

Figure 1.3: An illustration of Theorems 1.9 and 1.10.

Copnumbers of graphs have also been considered by Beraducci and Intrigila [3] using retracts. The first result is similar to Theorem 1.1.

Theorem 1.11 (Beraducci and Intrigila [3]) *Let G be a graph and let H be a retract of G. Then* $c(H) \leq c(G)$.

Proof. Let H be a retract of the graph G. and let f be a retraction map from G onto H. Now $c(G)$ cops have a winning strategy on G. Through the map f. the cops are able to translate this winning strategy onto H. Hence the copnumber of H is at most $c(G)$. □

Beraducci and Intrigila pursued the idea used in the previous theorem. and proved the next theorem along with a corollary.

Theorem 1.12 (Beraducci and Intrigila [3]**)** *Let G be a graph and let H be a retract of G. Suppose $c(H)$ cops are playing on H. After a finite number of moves. these cops can say with certainty that the robber will be apprehended if he moves onto H.*

Proof. Suppose $c(H)$ cops are playing on a retract H of the graph G. Through the retraction map f. the cops can consider the robber's movements on G as being made on H. After a finite number of moves. the $c(H)$ cops are able to capture the robber's image on H. Once this has been accomplished. one of the cops moves so as to stay with the image of the robber. Since f is the identity function on H. should the robber decide to move onto H. he would be immediately apprehended. □

Corollary 1.1 (Beraducci and Intrigila [3]**)** *Let G be a graph and let H be a retract of G. Then $c(G) \leq max\{c(H). c(G \setminus H) + 1\}$.*

Proof. Suppose the robber's image has been apprehended on H as discussed in the proof of Theorem 1.12. One cop is needed to stay with this image and therefore to prevent the robber from moving onto H. The remaining $c(H) - 1$ cops can move onto $G \setminus H$ and aid in capturing the robber there. Now if $c(H) - 1 > c(G \setminus H)$ then $c(H)$ cops are able to capture the robber on G. Otherwise. $c(G \setminus H)$ are needed to capture the robber on $G \setminus H$ and one more cop is needed to stay with the robber's image on H for a total of $c(G \setminus H) + 1$. Since $c(G \setminus H) \geq c(H) - 1$ or equivalently $c(G \setminus H) + 1 \geq c(H)$. this number of cops is also able to apprehend the image of the robber on H. Hence the copnumber of G is at most $\max\{c(H). c(G \setminus H) + 1\}$. □

It has been shown in Theorem 1.4 that the strong product of a finite collection of copwin graphs is also copwin. An analogous result for the copnumbers of graphs due to Neufeld and Nowakowski [9] is included next. It bounds the copnumber of the strong product of two graphs in terms of the individual copnumbers of these graphs.

Theorem 1.13 (Neufeld and Nowakowski [9]) *Let G and H be graphs. Then $c(G \boxtimes H) \leq c(G) + c(H) - 1$.*

Proof. Let G and H be graphs and consider the strong product $G \boxtimes H$. Let h be the projection map from $G \boxtimes H$ onto $\{x\} \cdot H$, the subgraph of $G \boxtimes H$ whose first coordinate is $x \in G$, and let g be the projection map from $G \boxtimes H$ onto G. Note that $\{x\} \cdot H$ can be thought of as a copy of H in the product. Now $c(H)$ cops are needed to capture the image of the robber on $\{x\} \cdot H$. One of these cops is then needed to remain with the image of the robber. This is known as shadowing the robber. The remaining $c(H) - 1$ cops, along with one additional cop, are available to capture the image of the robber another time. This process is repeated until $c(G)$ cops, $\{s_1, s_2, \ldots s_{c(G)}\}$, are shadowing the robber. In addition to these $c(G)$ cops, there are $c(H) - 1$ other cops who have been playing. These $c(H) - 1$ cops have participated in capturing the robber $c(G)$ times. Now these $c(G) + c(H) - 1$ cops have a winning strategy on the strong product. The cops $s_1, s_2, \ldots, s_{c(G)}$ shadow the robber and play their winning strategy on G. Hence $c(G \boxtimes H) \leq c(G) + c(H) - 1$. □

1.2 Searching

It has been noted previously that in the game of Cops and Robber, both sides have perfect information. Suppose the game is modified so that the cops have no information about the position of the robber. This version of the game is known as **searching** and was introduced by T. D. Parsons [11]. This game is usually formulated in terms of apprehending an infinitely fast robber, or clearing an area (graph) of airborne contaminants. In keeping with the remainder of the thesis, we will think of this game in terms of the robber. To differentiate this infinitely fast robber from the robber we have been considering up to now, this robber will be referred to as an f-robber. We note that the f-robber can be located on edges of the graph. This is because the original idea of the game was to search for people in caverns.

Given a graph G, the main objective arising out of the searching game is to be able to determine the minimum number of searchers required to apprehend the f-robber. This number will be referred to as the **search number** of G and is denoted $s(G)$. Suppose $s(G) = k$. The k searchers have an efficient strategy or way of searching the graph for the f-robber. Define searcher i's strategy as the path he follows on the graph G. It will be useful to think of this strategy as a continuous function $f_i : [0, \infty) \rightarrow G$ where $f_i(t)$ is the position of the ith searcher at time t. The set $\{f_i : 1 \leq i \leq k\}$ is the **collective strategy** of the k cops. Similarly, define the f-robber's position at time t as $e(t)$. Clearly the search is over when $f_i(t_0) = e(t_0)$ for some $i \in \{1, 2, \ldots, k\}$ and some $t_0 \in [0, \infty)$. Here the search number can be thought of as the minimum cardinality of all such collective strategies.

We begin with an intuitive and useful result that bounds the search number of a subgraph in terms of the search number of the larger graph.

Theorem 1.14 (Parsons [11]) *Let G be a graph and let H be a subgraph of G. Further, let H be connected. Then $s(H) \leq s(G)$.*

Proof. Let G be a graph. The $s(G)$ searchers have a strategy for searching G. This strategy can be modified and used to search H by $s(G)$ searchers. The searchers follow the strategy for G except that they disregard those parts of the strategy that indicate they move outside of H. In terms of functions, let the strategy used to search G be given by $\{f_i : 1 \leq i \leq s(G)\}$. Define continuous functions $h_i : [0, \infty) \rightarrow H$ such that $\forall f_i(t) \in H$, $h_i(t) = f_i(t)$. The function $h_i(t)$ is simply the restriction of $f_i(t)$ to values of t for which $f_i(t)$ is in H. Such functions are desirable because the graph H is being searched rather than the graph G. Now $h_i(t)$ is the strategy used on H by the ith cop, and so $\{h_i : 1 \leq i \leq s(G)\}$ defines the desired strategy on H used by $s(G)$ cops. Hence $s(G)$ cops have a strategy for searching H, and so $s(H) \leq s(G)$. □

We note that this relationship does not hold for copnumbers: that is, if G is a graph and H is a subgraph of G, it is not true that $c(H) \leq c(G)$. To see this, consider the graphs G and H shown in Figure 1.4. The graph G has copnumber 1 since the middle vertex is adjacent to all of the other vertices. The graph H has copnumber 2. Hence $c(H) \not\leq c(G)$.

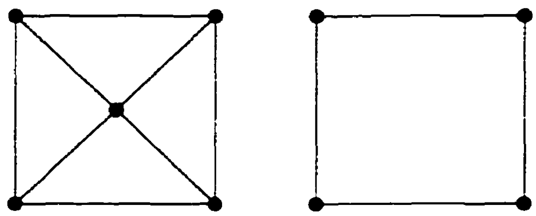

Figure 1.4: $2 = c(H) \not\leq c(G) = 1$.

It is straightforward to obtain an upper bound for the required number of searchers of any graph. Suppose the graph G_n has n vertices. Then n searchers can position themselves on the vertices of G_n, one searcher to a vertex. One additional searcher is needed to search the edges of G_n. Hence $s(G_n) \leq n + 1$.

Now let u be a vertex of a tree G. A **branch** of G is defined to be a maximal subtree such that u has degree one in the subtree. This definition is needed in the next example.

Example: Consider the graph, G_{25} shown in Figure 1.5.

Clearly two cops are sufficient to search this graph. One cop remains stationary on the root (indicated by a double circle) while the second searches each of the branches of the tree. The stationary cop prevents the f-robber from moving into a previously searched area. The upper bound obtained previously gives $s(G_{25}) \leq 26$. Clearly it is often possible to do much better than that which is suggested by this particular upper bound.

Although particular searching strategies are not considered in depth here, it should

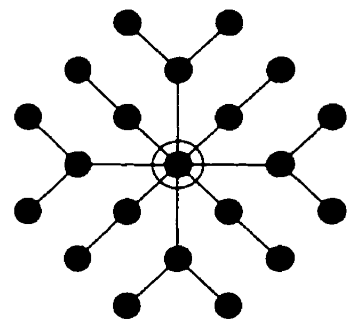

Figure 1.5: G_{25}

be noted that this example can help provide insight into the kinds of strategies that are needed by the searchers. The example indicates that the search number of a graph depends upon the searchers being able to prevent the f-robber from moving into an area that has already been searched. In terms of airborne contaminants, recontamination must be prevented.

It is clear that it is desirable to improve upon the first upper bound on the search number of a graph that has been presented here. The definitions presented next will aid in improving this upper bound.

Definition 1.12 *Let G be a graph. A **linear layout** of G is the graph G drawn so that all of its vertices lie in a straight line. This line is assumed to be horizontal.*

Consider a linear layout of a graph G, and imagine a vertical line dividing the layout into two pieces. There will be a number (possibly zero) of vertices on the left side of this line which are adjacent to vertices on the right. This idea is needed for the next definition.

Definition 1.13 *Let $L = \{x_1, x_2, \ldots, x_n\}$ be a linear layout of a graph G. Let n_i be the number of vertices in $\{x_1, x_2, \ldots, x_i\}$ which are adjacent to some vertex in $\{x_{i+1}, x_{i+2}, \ldots, x_n\}$. The vertex separation number of L is $vs(G, L) = max\{n_i : i = 1, 2, \ldots, n-1\}$. The **vertex separation number** of G is $vs(G, L) = min\{vs(G, L) : L$ is a linear layout of $G\}$.*

The vertex separation number of a graph G is used to place a bound on the search number of a graph as indicated in the next theorem. The proof of the theorem does not make full use of the dynamic nature of the searchers. The technique used will appear again in the proof of Theorem 3.5 with the cops replaced by static objects known as traps.

Theorem 1.15 (Ellis, Sudborough, and Turner [5]) *Let G be a graph with search number $s(G)$ and vertex separation number $vs(G)$. Then $s(G) \leq vs(G) + 1$.*

Proof. Suppose the game is taking place on a graph G and the f-robber is playing against $vs(G) + 1$ cops. We will show that these cops have a winning strategy.

Let's represent G as a linear layout on which $vs(G)$ is realized. Further, let's imagine a vertical line which will move across the layout from the left.

Now $vs(G)$ cops position themselves on vertices to the left of the vertical line in such a way that no unoccupied vertex to the left of this line is adjacent to a vertex on the right. This is possible by the definition of vertex separation number.

The remaining cop positions himself on the vertex immediately to the right of the vertical line. Now suppose that the line moves to the right, passing only one vertex. The $vs(G) + 1$ cops are now all positioned to the left of the line.

Consider a vertex v to the right of the line, and suppose this vertex is adjacent to one or more vertices on the left. These vertices adjacent to v must be occupied by cops. This is because v can only be adjacent to the vertex that was previously immediately to the right of the line (and is now immediately to the left) which is occupied by a cop, or a vertex to the left of both the present and previous vertical lines. These vertices are all occupied by cops.

There are $vs(G) + 1$ cops positioned to the left of the line, but only $vs(G)$ are needed to prevent the f-robber from crossing this line. So one of the cops is extraneous, and can move and position himself immediately to the right of the new vertical line. This process repeats until the vertical line immediately preceeds the rightmost vertex. This is the vertex on which the f-robber must be located. When the extraneous cop crosses the line, the f-robber will be apprehended.

Hence $vs(G) + 1$ is an upper bound for the number of cops needed to apprehend the f-robber. □

At this time, it should be noted that this result also places a bound on the copnumber of a graph G. This is because the games of Cops and Robber and searching differ in the amount of information available to the cops. Surely cops with information concerning the robber's position will be able to apprehend the robber at least as efficiently as cops with no information, and therefore $c(G) \leq s(G)$. Hence, $vs(G) + 1$ is also an upper bound for the copnumber of a graph. This result is stated as a corollary.

Corollary 1.2 *Let G be a graph with copnumber $c(G)$ and vertex separation number $vs(G)$. Then $c(G) \leq vs(G) + 1$.*

The remaining two theorems that are presented in this section are due to T. D. Parsons [11]. The first places a lower bound on the search numbers of trees. The latter enables us to characterize trees with search number n for all n, which is in sharp contrast to the copnumber for trees.

Theorem 1.16 (Parsons [11]) *Let n be positive and let G be a tree. Then $s(G) \geq n+1$ if and only if G has a vertex u where there are at least three branches, B_1, B_2, B_3 satisfying $s(B_j) \geq n$ for $j = 1, 2, 3$.*

The characterization mentioned previously involves a recursion of sets $T_1, T_2, \ldots$ of trees. Let T_1 be composed of two vertices joined by an edge. Suppose T_n has been defined. The construction presented below allows us to define T_{n+1}. One member from each of the isomorphism classes of trees resulting from the construction is included.

To begin the construction, three trees must be chosen from T_n which has already been defined. These trees will be denoted T_1, T_2, T_3. These trees should be disjoint although not necessarily distinct in the sense of isomorphism.

Now a vertex should be chosen from each of the three trees. A chosen vertex must be one of two types. If a vertex of degree one is not chosen, then the vertex should not be adjacent to a vertex of degree one. Let the vertex chosen from T_i be denoted u_i.

Next, a fourth vertex must be chosen. This vertex may not be a member of the vertex set of either T_1, T_2, or T_3. This vertex is denoted v.

Finally, a new tree is constructed. The vertex v will be the root of this tree. If vertex u_i has degree one in T_i then this vertex becomes vertex v in the new tree. Otherwise, u_i is joined to v by an edge.

We proceed with an example.

Example:

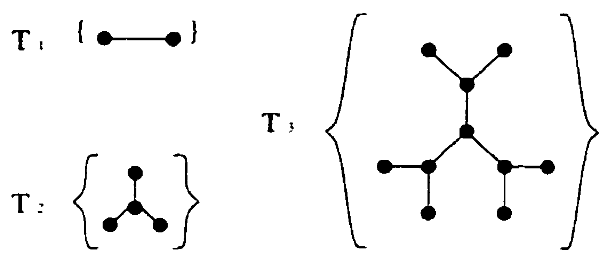

Figure 1.6: T_1, T_2, and T_3, the first three sets.

The characterization using this construction is made precise in the next theorem.

Theorem 1.17 (Parsons [11]) *Let $n \geq 2$ and let G be a tree. Then $s(G) = n$ if and only if G has a subtree homeomorphic to a tree in T_n, but has no subtree homeomorphic to a tree in T_{n+1}.*

We introduce a variation of this game involving the use of photo radar in the next chapter.

1.3 Watchman Problem

In this final section of the chapter, a problem similar to those presented so far is considered. This problem is known as the Watchman Problem, and was introduced by Hartnell, Rall, and Whitehead [7].

Suppose a company is providing security for a business. Important locations on the property can be thought of as vertices in a graph, and the security company must monitor these vertices. A vertex v is considered to be monitored if a watchman either visits v or a neighbor of v from which v can be viewed.

The first definition presented here is an extension of the idea of a dominating vertex as defined in Section 1.1.2.

Definition 1.14 *Let G be a graph. A set D of vertices of G is said to be a* ***dominating set*** *if $\forall v \in V(G) \setminus D$, $\exists d \in D$ such that v and d are adjacent.*

A walk W is said to be a **dominating walk** if its vertices are a dominating set of G. A walk with minimum length is said to be an **optimal watchman's walk**. The length of such a walk is denoted $W_1(G)$.

Suppose only a single watchman is available. This watchman will want to minimize the number of steps required to monitor the vertices of G, the underlying graph. Equivalently, the watchman would like to determine a closed walk of minimum length whose vertex set dominates G; that is, he would like to determine an optimal watchman's walk.

Example: Consider the graph G shown below. The vertices of G are numbered to indicate an optimal watchman's walk.

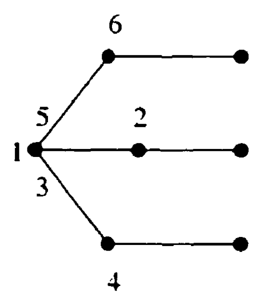

Figure 1.7: An optimal watchman's walk of length 6.

Suppose the business for which security is being provided requires that every vertex is monitored at least once every k units of time, where k has been previously agreed upon. The security company will want to minimize the number of watchmen it provides. It must also determine the walks followed by these watchmen as they make their rounds.

Example: Consider the graph G shown in Figure 1.7. Two watchmen are able to monitor the vertices of G every three seconds. One watchman begins at the vertex labeled 1 and follows the vertices in the order indicated by the labels. Once vertex 6 is visited, the watchman returns to vertex 1 and continues to visit the vertices in the order indicated by the labels. Similarly, the second watchman begins at the vertex labeled 3 and follows the vertices in the order indicated by the labels. Vertex 1 follows vertex 6. In this way, each of the vertices is monitored once every three seconds.

Consider a graph G. Hartnell, Rall, and Whitehead (private communication, 1999) have proven that if $W_1(G) = k$, then two cops are able to reduce the time needed to monitor the vertices of G only by half (except in certain special cases when the time can be reduced by one additional unit). As shown in the previous example, one of the watchmen follows an optimal watchman's walk. Let the ith vertex visited by the watchman on this walk be denoted i. The second watchman follows the same walk as the first, but begins on vertex $\lceil k/2 \rceil$. In this way, the vertices of G are monitored once every $\lceil k/2 \rceil$ units of time.

Hartnell, Rall, and Whitehead [7] have shown that if we are given a graph G and a positive integer p, the problem of determining if $W_1(G) \leq p$ is NP-complete since it can be reduced to the Hamiltonian cycle problem.

1.4 Open Question

It is known from the Copwin Strategy that at most n moves are required to catch a robber on an n vertex copwin graph. How many cops are required to catch the robber in time T? (There would be a statute of limitations or some fixed resource, such as gas, for the chase.) In general, we would like to answer this question for an arbitrary graph G.

Chapter 2

Special Photo Surveillance

In this chapter. several similarly themed variations of the game of Cops and Robber are introduced. In the original version of the game. the opponents have perfect information. In the versions introduced here. it is proposed that the cops can only get information about the robber's position through the use of special photo surveillance equipment. As in the original game. it is assumed that the robber has perfect information.

2.1 Photo Radar

The first variation of Cops and Robber that will be introduced is the use of photo radar by the cops. Suppose the game is being played on a graph G. Photo radar units are placed on the edges of G. These units alert the cops if the robber moves along an edge equipped with a photo radar unit. The units also indicate the direction in which the robber is moving. The minimum number of photo radar units required by a single cop to guarantee the capture of the robber on a graph G will be referred to as the **photo radar number** of G. and will be denoted $pr(G)$.

Lemma 2.1 *Given a single cop and a finite number n of photo radar units. there exists a star on which the cop cannot guarantee the capture of the robber using only* *these $n \geq 1$ units.*

Proof. Consider the star $K_{1,k}$. To show that no bound can be placed on the number of photo radar units needed, it will suffice to show that $k - 3$ photo radar units will not ensure that the robber is caught.

Suppose $k - 3$ photo radar units are placed on the edges of the star described above, one photo radar unit per edge. This leaves three edges with no surveillance, and it is possible for the robber to evade the cop. To see this, suppose the cop is at the center of the star and the robber is located on a leaf incident with an edge that has no photo radar. (Otherwise, the cop will move to the center during his next turn.) The probability of the cop choosing the correct leaf at this time depends on the leaf from which he has just come. If he has just moved from one of the leaves incident with an edge with no photo radar, he knows the robber must be located on one of the remaining two such leaves. Therefore, the probability of catching the robber on the next move is 1/2. Otherwise, the robber is located at one of three possible vertices and so the probability of catching him is 1/3. Now while the cop is investigating a particular leaf (we assume this is not the leaf on which the robber is hiding) the robber is able to move to the center. By the time the cop is able to return to the center, the robber has been able to move to a leaf. Since the cop knows the robber must be located on one of the two leaves he did not check, his probability of catching the robber during the next move is 1/2. There is a nonzero probability that this process will continue indefinitely with the result that the robber is not caught and wins the game.

Hence $k - 3$ photo radar units will not ensure that the robber is caught. Therefore we can say that no absolute bound can be placed on the number of photo radar units needed to catch the robber for members of the family of stars. □

Theorem 2.1 *Given a single cop and a finite number n of photo radar units, there exists a tree on which the cop cannot guarantee the capture of the robber using only these $n \geq 1$ units.*

Proof. To prove this theorem, we will show that playing this game on a tree can be reduced to playing on a star, the situation examined in the previous lemma.

Suppose the game is being played on a tree, T. The robber's initial position can be thought of as a vertex of a subgraph S of T, where S is a star. Now if both the cop and the robber move only within S, we are done. Otherwise, suppose the cop moves outside of S. The robber passes on each of his moves until the cop enters S. Note that if the cop previously made a move within S, then the cop must return to S by the same vertex from which he left. This is because of the acyclic nature of trees. Hence the cop's moves outside of S have no impact on the game, and so we can consider the game as if it were only being played on the vertices of S.

Hence by the previous lemma, if we consider members of the family of trees and only one cop is available, then no bound can be placed on the number of photo radar units required to catch the robber. □

This situation is very similar to that of searching a tree.

Consider a tree T_n with n vertices. It is useful to know how much information regarding the robber is needed to ensure his capture on this tree. Now $pr(T_n) \leq n-1$ since the photo radar units can be placed one to an edge. If the robber doesn't move, the game is equivalent to the searching problem. If the robber moves, the game is equivalent to Cop and Robber. In either case, the cop is able to apprehend the robber. Hence $n-1$ photo radar units guarantee the capture of the robber on T_n by a single cop. The next theorem places a lower bound on the number of photo radar units needed. We conjecture that the inequality in this theorem is actually an equality.

Theorem 2.2 *Let T be a tree, and let k be the minimum number of edges of T whose removal leaves no vertices of degree ≥ 3. Then $pr(T) \geq k$.*

Proof. Let k be the minimum number of edges of T whose removal leaves no vertices of degree ≥ 3. Suppose only k' photo radar units are available, $k' < k$. Then there is at least one vertex in T incident with at least three edges without photo radar. From the proof of Lemma 2.1, we know that the robber is able to evade the cop. Therefore, $pr(T) \geq k$. □

It seems that photo radar units are not powerful in the sense that a large number of units are needed by a cop to guarantee the robber's capture. In the third chapter. a more powerful aide is made available to the cop: that is. the cop is able to use traps in his search for the robber.

2.2 Streets and Avenues

In the version of the game Cops and Robber introduced in this section. it is proposed that the cop only knows the position of the robber if the robber appears on a line of the grid that is under surveillance. The surveillance allows the cop to monitor all of the vertices along the grid line that is equipped with the surveillance. Note that vertical grid lines will be referred to as **avenues**. while horizontal grid lines will be referred to as **streets**. To begin. several of the strategies that will be used throughout this section are described in detail. These strategies are a combination of strategies used in Cops and Robber and searching. They are described under the assumption that all streets and avenues are under surveillance: that is. the cops have perfect information. To show that these strategies will result in the capture of the robber. some **measure** is needed to indicate that. after a finite number of moves. a cop will arrive at the vertex occupied by the robber. In the strategies that follow. this measure will be the vertical and horizontal distances between the cop(s) and the robber. Once both distances have been decreased to zero. the robber is apprehended. The highlights of the strategies are presented in italics.

Strategy: Blocking (One Cop)

Consider a rectangular grid and assume that both the cop and the robber have perfect information. The strategy described here prevents the robber from moving in one of the four possible directions. The case where the robber is prevented from moving upwards to another street on the grid is described in detail. The other three cases are similar.

Firstly. the cop moves along an avenue until he is positioned one street above the *robber: that is, he moves along an avenue until the vertical distance is one.* This is possible, as explained below, because there are only a finite number of streets. If the robber chooses to move along a street or to pass, the cop is able to decrease the vertical distance between himself and the robber. If the robber chooses to move along an avenue, he will eventually reach the top (bottom) of the grid. At this point, he can choose to move along this street or he can reverse direction and begin moving down (up) an avenue. In both cases, the cop will be able to decrease the vertical distance between them. After a finite number number of moves, the cop will be exactly one street above the robber. (There is one problematic case. If the robber has reached the topmost street and decides not to move downward but rather to move along this street, the cop will not be able to move above him. However the cop will also reach this street and will be able to apprehend the robber. This is because the robber has chosen not to leave this street and the vertical distance has been decreased to zero. The street has only a finite number of vertices and so, after at most $n - 1$ moves, the cop is able to decrease the horizontal distance to zero as well.)

It should be noted here that once the cop has attained his position on the street above the robber, he is unable to drop down to the same street as the robber as this would give the robber the opportunity to move up one street. Also, the cop must move down one street whenever the robber chooses to do so. This prevents the robber from increasing the vertical distance.

Secondly, the cop must move so that he is either positioned directly over the robber, or he is positioned diagonally opposite to the robber across one of the two blocks in which the position of the robber is one of the bottom corners as shown in Figure 2.1.

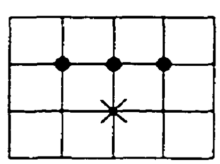

Figure 2.1: The robber is indicated by an x. The shaded circles indicate the three possible positions of the cop which prevent the robber from moving upward one street.

Again this is possible, as explained in the remainder of the paragraph, because there are only a finite number of avenues. If the robber moves along the street toward the cop, the horizontal distance is decreased. If the robber only moves along the street away from the cop, he will eventually reach the edge and then the cop will decrease the horizontal distance. If the robber moves onto the same street, again the cop can decrease the horizontal distance. When the robber finally moves off the street, so that the vertical distance is again one, the cop moves so as to reduce the horizontal distance by a further one, and then moves to maintain the vertical distance of one. One of these situations must occur at least every $n + m - 1$ moves.

We can assume that eventually this positioning of the cop prevents the robber from moving upward. Otherwise, turn the board around. If the robber did attempt to move up a street, he would move onto a vertex either occupied by or adjacent to a cop resulting in a win for the cop.

Strategy: Blocking (Two Cops)

Again, consider a rectangular grid and assume that both the cop and the robber have perfect information. This strategy results in the robber being backed into a "corner", and hence a win for the cops. (It should be noted that this is not the same corner as was described in Section 1.1.2.)

The cops each follow the blocking strategy for a single cop, one either above or below the robber, the other to the left or right of the robber. It should be noted that the presence of two cops restricts the robber's movements even before the cops come within one street or avenue of the robber. This is because the robber will not want to decrease the horizontal distance between himself and the cop to his left or right, and he will not want to decrease the vertical distance between himself and the cop above or below him. *The robber will be forced into one of the four corners depending on the locations of the two cops.* Hence a win for the cops will result.

Strategy: Blockading (Two Cops)

The cops initially ignore the movements of the robber until they are positioned on

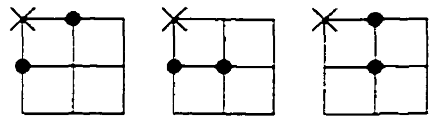

Figure 2.2: The three possible ways the robber is forced into the upper left 'corner'. Both the vertical and horizontal distances between the cops and the robber have been decreased to one. The cops win on the next move.

adjacent vertices of the $m \times n$ grid. In the description of the blockading strategy that follows, the cops are applying the strategy from above the robber. Hence it is assumed that the cops are located on adjacent vertices of the same street, and that this street is above the one on which the robber is located. However, this strategy can be applied regardless of the initial positions of the cops and the robber.

The cops proceed to move toward the robber along the street on which they are located. Because there are a finite number of avenues before the edge, the cops are able to decrease the horizontal distance and *position themselves on this street so that one cop is directly above the robber. The second cop is adjacent to the first.* Now it is to the advantage of the cops if the robber chooses to move toward the cops and decrease the vertical distance between them. Similarly, *if the robber chooses to pass, the cops are able to move closer to him by one street. If the robber moves downward along an avenue, the cops also drop down to the next street* to prevent the vertical distance from increasing. *When the robber moves horizontally along a street, the cops must move so that one of the cops is on the same avenue as the robber and the other cop is adjacent to the first.* It should be noted that both cops remain on the same street. *Maintaining these positions relative to the robber as the robber moves along a street sometimes forces the cops to move horizontally along the street in the same direction as the robber. However, sometimes no horizontal movement is needed by the cops and they are able to use their move to drop down a street and decrease the vertical distance between themselves and the robber.* To see this, suppose one cop is on the same avenue as the robber and the other cop is to the right of the first. If the robber moves to the left, the cops must also move to the left. However, if the robber moves to the right, the second cop is then on the same avenue as the robber and the first is to the left of the second. Hence no horizontal movement is needed by the cops, and they take the opportunity to move down one street.

Since the grid is finite, *the vertical distance between the cops and the robber will eventually be decreased to one street.* (This is because the robber will eventually reach the bottom of the grid. He is forced to drop down one street at least once every $n-1$ moves. Once the robber has reached the bottom of the grid, he cannot increase the vertical distance between himself and the cops and can only maintain the current distance by moving horizontally. However after every $n-1$ moves, he will be forced to change direction and the cops are able to decrease the vertical distance by one.) *At this time, the movements of the robber are further restricted. It is known that one cop is directly above the robber. It will be assumed for the purpose of discussion that the second cop is to the right of the first.* The other case is similar. *Now in addition to being unable to move upward or pass, the robber is unable to move to the right.* If he did, the second cop would capture him on the cops' next move.

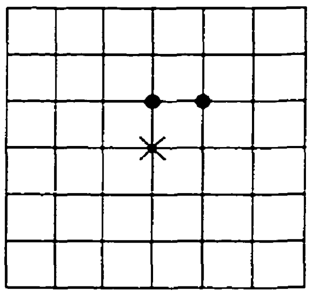

Figure 2.3: Blockading strategy. The robber is indicated by an x and the cops are indicated by shaded circles. The robber cannot move upward or to the right and avoid capture.

Hence the robber is forced to move downward or to the left. As the robber moves, the cops move so as to maintain their positions directly above the robber, and diagonally opposite and to the right of the robber. This forces the robber to continue to move downward and to the left. Since the grid is finite, the vertical and horizontal distances will decrease to one. and *the robber will be trapped in the bottom left corner of the grid after a finite number of moves.* He will be apprehended by the cops on the next move.

Strategy: Blockading (k Cops. $k > 2$)

Blockading on a rectangular $m \times n$ grid with k cops is similar to blockading as previously described with two cops. In this case. *the robber will be forced onto one of the streets or avenues that form the perimeter of the grid where he will be apprehended.* It should be noted that blockading from the right will be described here. The other cases are similar.

The cops align themselves on k consecutive vertices along an avenue to the right of the robber. They will not begin moving horizontally toward the robber until one of the cops is located on the same street as the robber.

If the robber chooses to pass or move to the right. the cops are able to decrease the horizontal distance between themselves and the robber. If he chooses to move to the left. then the cops will also move to the left. They will not let the robber increase his horizontal distance from them. *If the robber chooses to move upward (downward) along an avenue and the uppermost (lowermost) cop is on the same street as the robber. the cops will have to move vertically in the same direction as the robber. Otherwise. the cops move toward the robber.* In this way. the cops are able to decrease the horizontal distance by $k - 1$ every $m - 1$ moves.

Since there are a finite number of avenues before the edge. *the cops will be able to decrease the distance between themselves and the robber to one avenue. Also. the robber will be forced onto the leftmost avenue after a finite number of moves. If there are cops both on the street above the robber and on the street below the robber. the cops will win on the next move. Otherwise. the upper or lowermost cop is on the same street as the robber. Suppose it is the uppermost cop.* The other case is similar. *The robber is able to move upward. The cops also move upward.* Since the number of streets is finite. *the robber will be forced into the upper left corner after at most $m - k$ moves.* The cops will capture him on their next move.

It should be noted that capture occurs faster with blockading when there are k cops rather than two. This is because the k cops are able to decrease the distance between themselves and the robber more quickly.

Theorem 2.3 *If k consecutive streets are not under surveillance, then k cops are necessary and sufficient to capture the robber.*

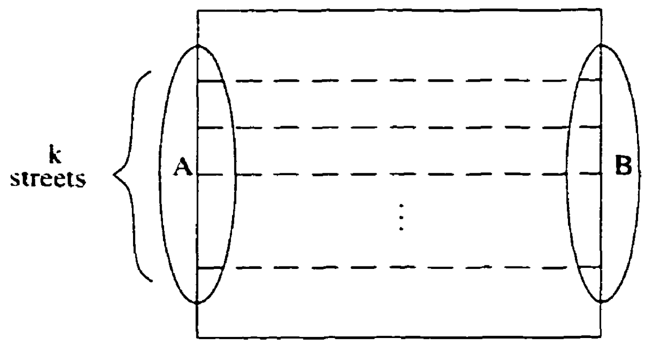

Figure 2.4: A grid with k consecutive streets not under surveillance.

Proof. Suppose k consecutive streets are not under surveillance. We wish to show that k cops are necessary and sufficient to capture the robber on such a grid.

Suppose the robber does not appear on the surveillance; that is, the robber is hiding in the group of unwatched streets. Since there are as many cops as streets, they align themselves at one end, say the right, of the group of unwatched streets, one to a street. Each cop moves along one of the unmarked streets. The k cops are always located on the same avenue. Since there is a cop for every street, the robber cannot slip by the cops as they are sweeping this section of streets.

If the robber is not apprehended during this sweeping, he is forced to move onto a street that is under surveillance. This street will be either directly above or below the unwatched streets. Since the cops are sweeping from the right, the robber must appear to the left of the cops.

The cops each make a move along the avenue in the direction of the robber. The avenue is the one on which they are located when the robber appears on the surveillance. They then adopt the blockading strategy. Since the robber is at most one street above or below the group of cops, he is unable to slip behind them into the previously searched portion of the group of unwatched streets. Hence if the robber moves back onto this group of unwatched streets, the cops resume the sweeping strategy. After a finite number of moves, the robber is apprehended. Hence if k consecutive streets are not under surveillance, then k cops will suffice to apprehend the robber.

Suppose there are only $k-1$ cops available for the search. There exist k paths between A and B as shown in Figure 2.4. The $k-1$ searchers can only occupy $k-1$ of them. This leaves one path on which the robber can move and evade the cops. Hence if k consecutive streets are not under surveillance, then k cops are necessary to apprehend the robber. □

Theorem 2.4 *If two nonconsecutive streets and one avenue are not under surveillance, then two cops are necessary and sufficient to catch the robber.*

Proof. Suppose the robber does not appear on the surveillance. The strategy used by the cops is to search the tree formed by the streets and the avenue that are not under surveillance. Cop 2 remains at the intersection of the avenue and the upper street while cop 1 moves upward along the avenue in the area labeled 1 in Figure 2.5. Cop 1 then returns along the same path until reaching cop 2 and the intersection. Similarly, cop 2 remains at the intersection while cop 1 searches area 2 and then area 3. Both cops then proceed down the avenue until they reach the intersection of the avenue with the second street. As before, cop 2 remains at the intersection while cop 1 searches areas 5, 6, and 7 in increasing order.

Now suppose that the robber appears ahead of the cops; that is, he appears below them on the grid. The cops do not change their strategy until one of them arrives at the same street as the robber. They then switch to the blockading strategy. The cops move in from the avenue with no surveillance forcing the robber to move away from this avenue. If the robber moves onto a street with no surveillance, the cops can deduce that he is either passing or moving away from them along that street.

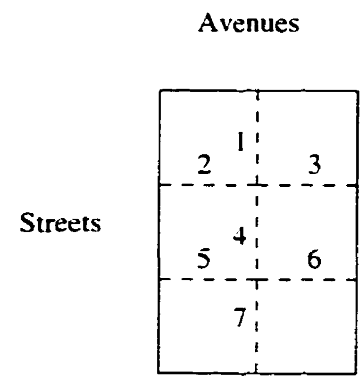

Figure 2.5: Two streets and one avenue not under surveillance.

In either case, they continue to move horizontally away from the avenue with no surveillance. No vertical movements are needed by the cops since the robber cannot move vertically without reappearing on the surveillance. Hence the cops are able to continue the blockading strategy even though the robber is not appearing on the surveillance. This blockading strategy decreases the distances between the cops and the robber, and will result in the robber being forced into a corner of the grid where he will be apprehended.

Suppose instead that the robber appears on the grid above the cops. It should be noted that this can only occur while a portion of one of the streets without surveillance is being searched, and that the robber can only appear one street above the cops. This is because if the robber is to appear suddenly on the surveillance, he must move onto the surveillance from one of the unwatched streets or from the unwatched avenue. Once portions of the streets or avenue have been searched, the robber cannot appear on streets or avenues adjacent to these searched portions. The cops then adopt the blocking strategy. Since the cop located at the intersection is exactly one street away from the robber, this cop can prevent the robber from moving onto the avenue with no surveillance. This effectively cuts the grid along the avenue, and reduces the area available for play in 'half'.

Without the avenue with no surveillance to consider. the streets with no surveillance are the only remaining troublesome places for the cops. If the robber moves onto a street with no surveillance. the second cop also moves onto this street on the next move. If the robber continues along this street. the cop can stay at most one move behind him. Since there are a finite number of vertices along the street. the robber must eventually move off of the street to avoid capture by the second cop. The first cop is able to monitor the robber's position while he is on the street with no surveillance. This is because the robber can only move in one direction (away from the second cop) and cannot pass and avoid capture. Hence. the blocking strategy continues when the robber reappears. After a finite number of moves. the cop will be forced into one of the upper corners. (The corner depends on which 'half' of the grid is being used for play.) The robber is then apprehended. Hence two cops are sufficient to apprehend the robber.

Clearly two cops are necessary to apprehend the robber. This is because one cop with perfect information would be unable to capture a robber on a grid. □

This result is generalized in the following theorem.

Theorem 2.5 *If k streets. no two of which are consecutive. and one avenue are not under surveillance. then two cops are necessary and sufficient to catch the robber.*

Proof. The proof of this theorem is nearly identical to that of the previous theorem. If the robber has not appeared on the surveillance. the cops search the tree formed by the streets and avenue with no surveillance. The search proceeds as indicated in Figure 2.6. If the robber appears below the cops on the grid. the cops adopt the blockading strategy once one of the cops reaches the same street as the robber. If the robber appears above the cops on the surveillance. the cops adopt the blocking strategy. In any case. the robber will be apprehended. Hence two cops are sufficient to capture the robber.

Clearly two cops are necessary to capture the robber. This is because one cop with perfect information would be unable to capture a robber on a grid. □

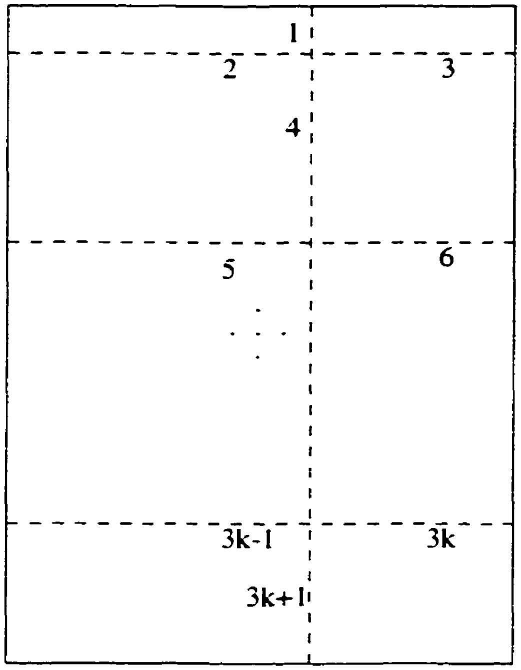

Figure 2.6: A grid with k streets, no two of which are consecutive, and one avenue not under surveillance (indicated by dashed lines).

Theorem 2.6 *If k consecutive streets and one avenue are not under surveillance, then $k+1$ cops are necessary and sufficient to catch the robber.*

Proof. Suppose k consecutive streets and one avenue are not under surveillance by the cops. It will be shown that $k+1$ cops have a winning strategy. To begin, k of the cops align themselves along the rightmost avenue, one cop on each of the streets with no surveillance. The $(k+1)$st cop waits at the intersection of the uppermost street with no surveillance and the avenue with no surveillance. The k cops move along the streets on which they began. The k cops are always located on the same avenue. It should be noted that k cops are required for this sweeping to prevent the robber from slipping by the cops into an area that has previously been searched.

Once the cops reach the avenue with no surveillance, they stop and wait while the $(k+1)$st cop searches that part of the avenue above the unwatched streets and then the part below the unwatched streets. The $(k+1)$st cop then positions himself at the intersection of this avenue with the lowermost street with no surveillance. The other k cops resume sweeping the unwatched streets. This sweeping of the unwatched streets will force the robber to move onto a street or avenue with surveillance after a finite number of moves.

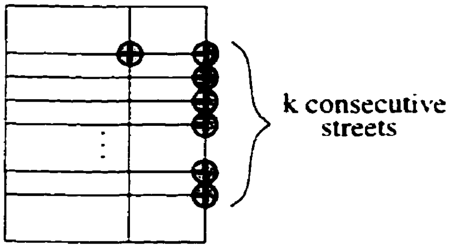

Figure 2.7: A game with k consecutive streets and one avenue not under surveillance. The shaded vertices indicate the initial positions of the cops.

Now suppose the robber appears below the cops on the grid. The cops begin moving down the avenue on which they were located when the robber appeared. Once the lowermost cop reaches the street on which the robber is located. the cops adopt the blockading strategy. If the robber disappears. the cops are able to move upward and reach the unmarked streets before the robber can slip into an area that was previously searched. This is because the uppermost cop never moves below the robber during the blockading strategy. The cops resume the searching strategy until the robber reappears. Since the robber can never slip behind the cops. this strategy leads to the capture of the robber. The case when the robber appears above the cops is similar except that the cops must move upward before beginning the blockading strategy. Hence $k + 1$ cops are sufficient to capture the robber.

Suppose there are only k cops available for the search. There exist k consecutive streets without surveillance. If the k searchers occupy all k of them. there will be no cop available to search the unwatched avenue. Otherwise. there is a street that is not occupied by a searcher on which the robber can move and evade the cops. Hence $k + 1$ cops are necessary to capture the robber. □

Theorem 2.7 *If two nonconsecutive streets and two nonconsecutive avenues are not* *under surveillance. then three cops are necessary and sufficient to catch the robber.*

Proof. The cops begin by searching the tree formed by the unwatched streets and avenues. The search proceeds in the order indicated in Figure 2.8.

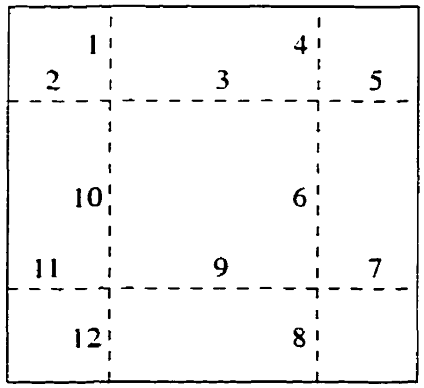

Figure 2.8: Two nonconsecutive streets and two nonconsecutive avenues not under surveillance.

One cop remains at the vertex representing the intersection of the upper street and the left avenue while a second cop searches areas 1 and 2. The cop on the intersection must remain there for the rest of the search to prevent the robber from moving onto a previously searched area. The second cop proceeds to search area 3 and then move to the intersection of the upper street with the right avenue. A third cop searches areas 4 and 5. These latter two cops then move down to the intersection of the lower street and the right avenue. One stays on the intersection while the other searches areas 7 and 8. They then both move to the last intersection. Again. one of the cops guards the intersection while the other searches areas 10. 11. and 12. This method of searching prevents the robber from slipping past the cops into an area that has already been searched. Hence if the robber appears on the surveillance and then disappears before the cops are able to apprehend him. the cops can simply resume the searching strategy.

Suppose the robber appears on a street below the cops. The cops will not change their strategy until one of the cops is on the same street as the robber. The other cop will be waiting on an intersection. The cops then adopt the blockading strategy which will result in the capture of the robber.

Suppose the robber appears on a street above the cops. The cop on the intersection is exactly one street below the robber and can block him from crossing this avenue with no surveillance. This cuts the grid along the avenue and restricts the area available for play. If the robber is on the side of the grid with no unwatched avenue. then two cops will suffice to catch the robber. The cop on the intersection is already using the blocking strategy to prevent the robber from crossing the unwatched avenue. The second cop also adopts the blocking strategy. and the capture of the robber results. We note that if the robber moves onto a street with no surveillance. the cops can assume that he is moving away from them along this street. and hence the blocking strategy can continue. If the robber is on the other side of the grid. then there are two nonconsecutive streets and one avenue with no surveillance. and there are two cops available for play. From Theorem 2.4. it is known that the cops have a winning strategy in this situation. Hence three cops are sufficient to capture the robber.

Two cops are not able to ensure that the robber is caught in this case. To see this. consider the cycle composed entirely of portions of streets and avenues not under surveillance. It is a 4-cycle. and it is known that two cops are necessary to capture a robber on a 4-cycle. Now while the cops are moving on this 4-cycle. the robber is able to hide in the other unwatched portions of the streets and avenues. If the cops choose to search these other areas and move off of the 4-cycle. the robber is able to evade the cops on the 4-cycle. Hence three cops are necessary to capture the robber.

□

This result is generalized in the following theorem.

Theorem 2.8 *If k streets. no two of which are consecutive. and two nonconsecutive avenues are not under surveillance. then $k+1$ cops suffice to catch the robber.*

Proof. The proof of this theorem proceeds similarly to that of the previous theorem. If the robber is not visible on the surveillance. the initial search occurs in the order

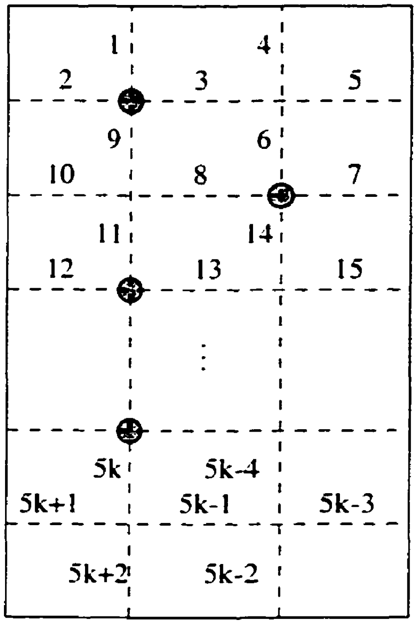

Figure 2.9: Two nonconsecutive avenues and k streets, no two of which are consecutive, not under surveillance.

indicated by Figure 2.9. The shaded circles represent vertices where a cop must remain once the search has reached that point. This prevents the robber from moving past the cops into an area that has been searched previously.

One cop is needed for each of the first $k - 1$ streets with no surveillance. Two additional cops are required to complete the search for a total of $k + 1$.

It should also be noted that Theorem 2.5 is used in the proof of this theorem where Theorem 2.4 was used previously. □

This result may not be the best possible. To see this, consider a cop c_i located on any position indicated by a shaded circle in Figure 2.9. Once a cop c_{i+1} is positioned on an intersection below this one on the same avenue, cop c_i is no longer required to maintain his position in order to ensure that the robber does not move into a previously searched area. However, the proof does not make use of this observation and so a better result may be possible.

Chapter 3

Cops and Robber with Traps

In this chapter, a variation of the game of Cops and Robber is considered. The cops have a number of traps at their disposal to aid in the apprehension of the robber on a graph G. The traps are placed on vertices of G. Once the robber moves onto a vertex with a trap, he is detained and the game is over with a win for the cops. Suppose a game is played with n cops and m traps. If the cops have a winning strategy, then G is referred to as (n,m)-win. Using this notation, a copwin graph is $(1,0)$-win. Our attention will be largely focused on $(1,m)$-win graphs. As in the original version, the game is played with perfect information.

Theorem 3.1 *For all n, there exists a graph G_n such that $c(G_n) = 2$ but one cop with n traps does not suffice.*

Proof. The graph $K_{n+2,n+2}$ is not copwin as there are many cycles present on which the robber can move and evade the cop. To show that the graph has a copnumber of 2, it must be shown that two cops have a winning strategy. Suppose the cops position themselves on one vertex in each set of the bipartition. This is a dominating set so the robber is either caught already or will be caught on the cops' first move.

The n traps can be placed on any of the $2n+4$ vertices. However, there is always a subgraph $K_{2,2}$ remaining which does not receive traps, and on which the robber can move and evade the cop.

Hence it has been shown that two cops cannot always be replaced by a single cop with a finite number of traps. □

Corollary 3.1 *Let G be a graph. Now $c(G) = 2$ is not equivalent to G being (1,1)-win.*

Theorem 3.2 *If R is a retract of a (1,1)-win graph G then R is also (1,1)-win.*

Proof. Suppose R is a retract of a (1,1)-win graph G, and let f be an edge-preserving map from G to R such that the restriction of f to R is the identity map on R. Since G is a (1,1)-win graph, the cop has a winning strategy on G. Through the map f, this strategy can be translated to R. The cop will play this winning strategy on R. Now the robber is playing on R. However, the cop will consider the robber's moves as being played on G even though they never take place outside the subgraph R. It should be noted that any edge used by the cop in G will be present in R since f is edge-preserving. Similarly, since f is the identity map on R, any edge used by the robber is also present. □

Corollary 3.2 *If R is a retract of an (n,m)-win graph G then R is also (n,m)-win.*

The proof of the corollary is omitted since it is identical to that of the main theorem. Now that it has been shown that the class of (1,1)-win graphs is closed under retracts, it is natural to ask if this class is a variety. Unfortunately, this is not the case. This is the subject of the next result.

Theorem 3.3 *The class of (1,1)-win graphs is not a variety.*

Proof. We show the class of (1,1)-win graphs is not a variety by showing that this class of graphs is not closed under products. Consider the product of a (1,1)-win graph and a copwin graph shown in Figure 3.1.

The resulting graph is not (1,1)-win. This will be shown by presenting a strategy that can be used by the robber to evade the cop. The vertices of the graph will be considered to be partitioned into three 4-cycles $\{(i,0),(i,1),(i,2),(i,3)\}$, $i = 1,2,3$.

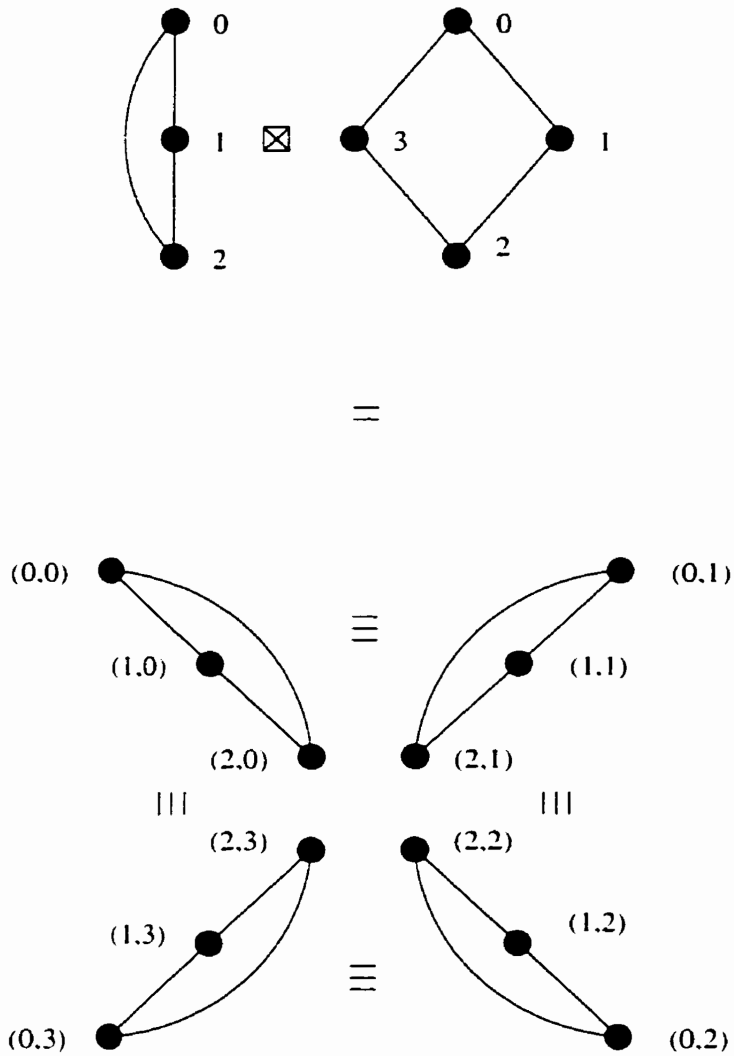

Figure 3.1: The strong product of a (1,1)-win graph (C_4) and a copwin graph (K_3). Three consecutive segments between adjacent subgraphs indicate that every vertex in one subgraph is adjacent to every vertex in the other.

This is simply to aid in the description of the strategy. As well, the vertex on each of these cycles that is farthest from the cop's position will be said to be diagonally opposite to the cop as shown in Figure 3.2.

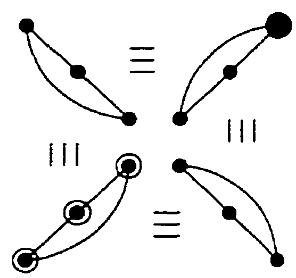

Figure 3.2: The vertices indicated by double circles are said to be diagonally opposite to the large shaded vertex. Again, three consecutive segments between adjacent subgraphs indicate that every vertex in one subgraph is adjacent to every vertex in the other.

The cop eventually lays the trap on a vertex in one of the three 4-cycles since the trap is not of use to the cop if it is carried during the entire game. The robber does not want to move onto the cycle that contains the trap. So the robber chooses a vertex in one of the remaining two 4-cycles. The robber will occupy the vertex on this cycle that is diagonally opposite to the position of the cop. As long as the cop stays on the same cycle, the robber also stays on the same cycle, moving so as to maintain his position diagonally opposite the cop. Similarly, if the cop moves to another cycle, the robber continues to move so as to remain diagonally opposite the cop, passing when necessary. In this way, the robber is able to evade the cop, and so this graph product is not (1,1)-win. □

A single cop playing on the graph product described in the previous theorem requires three traps to ensure the capture of the robber, one for each of the three 4-cycles described above. In general, a cop playing on the graph $K_n \boxtimes C_4$ requires n traps to win. Hence, it is not even possible to place a bound on the number of traps needed. Thus, we can conclude that $(1,n)$-win graphs do not form a variety for a fixed n.

The next result is an extension of Corollary 1.1.

Theorem 3.4 *Let G be an (n_1, m_1)-win graph, let H be a retract of G with $c(H) \leq n_0$, and let $G \setminus H$ be (n_2, m_2)-win. Then $m_1 \leq m_2$ and $n_1 \leq max\{n_0, n_2 + 1\}$.*

Proof. Suppose n_0 cops are playing on a retract H of G. Through the retraction map $f : G \rightarrow H$, the cops can consider the robber's movements on G as being made on H. After a finite number of moves, the n_0 cops are able to capture the robber's image on H. Once this is accomplished, one of the cops moves so as to stay with the image of the robber. Since f is the identity function on H, the robber is immediately apprehended if he ever moves onto H. Hence the robber is restricted to $G \setminus H$. It is known that n_2 cops and m_2 traps are required to win on $G \setminus H$. Hence $m_1 \leq m_2$.

There are $n_0 - 1$ cops available to move onto $G \setminus H$ and aid in capturing the robber there. Now if $n_0 - 1 > n_2$ then n_0 cops are able to capture the robber on G. Otherwise, n_2 are needed to capture the robber on $G \setminus H$ and one more cop is needed to stay with the robber's image on H for a total of $n_2 + 1$. Since $n_2 \geq n_0 - 1$ or equivalently $n_2 + 1 \geq n_0$, this number of cops is also able to apprehend the image of the robber on H. Hence the number of cops required on $G \setminus H$ is at most $\max\{n_0, n_2 + 1\}$. □

Theorem 3.5 *Let $vs(G)$ be the vertex separation number of a graph G. The number of traps needed by a single cop to search this graph is at most $vs(G) + 1$.*

Proof. Suppose that $vs(G) + 1$ traps are available for use by the cop. The linear layout which realizes $vs(G)$ is formed. Suppose there is a vertical line separating this linear layout into two pieces. Further suppose that this vertical line moves from left to right along the layout.

Suppose that all but one of the available traps are placed to the left of the vertical line. By the definition of vertex separation number, it is possible to place these traps on vertices such that no other vertex lying to the left of the line is adjacent to a vertex to the right.

At this stage, there is one additional trap that has not been utilized by the cop. This trap is placed on the first vertex to the right of the line. Now imagine moving the line to the right past a single vertex so that all $vs(G) + 1$ traps lie to the left of the line.

Now any vertex to the right of the vertical line can only be adjacent to vertices to the left that are occupied by traps. This is because any vertex to the right of the line is adjacent either to the vertex immediately to the left of the line (the vertex previously to the right of the old vertical line) which has a trap or to a vertex to the left of both the old and new vertical lines which has a trap.

Only $vs(G)$ traps are needed to prevent the robber from moving over the vertical line from the right. Since there are $vs(G) + 1$ traps to the left of the line, one of the traps is unnecessary and can be moved by the cop to the vertex immediately to the right of the new line. This process is repeated.

Therefore at most $vs(G) + 1$ traps are needed by a single cop to search the graph G. □

Definition 3.1 *Let G be a graph. Suppose G has isometric cycles of length three and four only; that is, every cycle of length greater than or equal to five has a shortcut. Then G is said to be an* ***H-graph***.

Definition 3.2 *A* ***handle*** *H of a graph G is composed of the vertices $X \cup \{b\}$ and is characterized by the following properties:*

1. *the vertex b is adjacent to at least one vertex $x \in X$.*

2. *there exists a vertex c which dominates $N[b]$ except possibly for some of X.*

3. *there exists a vertex a such that a is the bottleneck for X: that is, a is adjacent to $x \in X$, and it is known that any vertex that is adjacent to $x' \in X$ is also adjacent to c.*

4. *there exists a subgraph $Y \subseteq N(b) \subseteq X \cup Y \cup \{c\}$ such that $\forall y \in Y$, y dominates $x \in X$ except for a.*

5. a and c are adjacent.

A handle $X \cup \{b\}$ of an H-graph is shown in Figure 3.3.

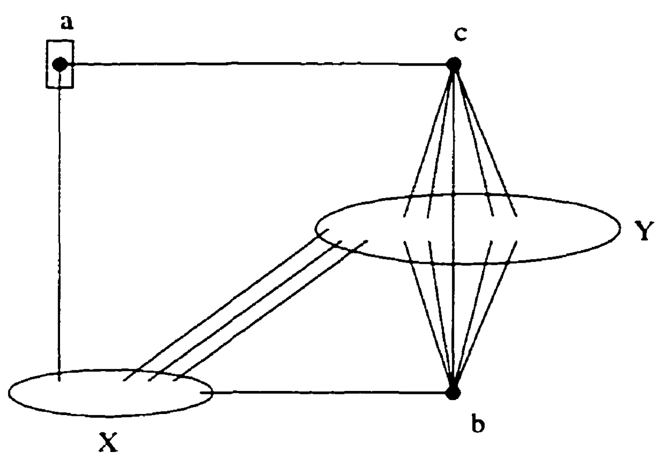

Figure 3.3: An H-graph. The trap is indicated by a rectangular box.

Theorem 3.6 *Let G be an H-graph and let H be a handle in G. If the robber is forced to move onto H, a win for the cop will result.*

Proof. Let G be an H-graph and let H be a handle in G as shown in Figure 3.3. Consider what occurs if the robber is forced to move onto H. Suppose the robber moves onto vertex b. If the cop is on c, then the cop wins on his next move. Suppose the cop moves onto c. The robber can pass and stay on b, move onto X, or move onto Y. If the robber passes, the cop wins on his next move since there is an edge between vertices b and c. If the robber moves onto Y, the cop also wins because c dominates $N[b]$ (except for some of X). If the robber moves onto X, the cop will move to a and lay the trap there. This will prevent the robber from being able to exit the handle through a. The cop then moves back onto c. Between these moves, the robber may have chosen to move back onto b or Y, or to remain in X, and it is the robber's move. After this move, the robber will be on a vertex in $\{c\} \cup \{b\} \cup Y \cup X$. If he is on c, the cop has won. If he is on b or $y \in Y$, the cop wins on the next move. Suppose the robber is on $x \in X$. The cop moves onto $y \in Y$. Since y dominates x, the cop wins on the next move. □

From this analysis, we conclude that the robber will not move onto the handle unless he is forced to do so. Hence it is not useful for the cop to play there unless the robber is forced to play there, in which case the cop has been shown to win. Therefore, we must determine if the cop can force the robber to move onto the handle. To do this, we remove this handle from the graph and play on the resulting subgraph. The next theorem tells us that this subgraph will be (1,1)-win if the original graph is (1,1)-win.

Theorem 3.7 *Let H be a subgraph of an H-graph G and let H be a handle. Let $G' = G \setminus H$. Then G' is (1,1)-win if G is (1,1)-win.*

Proof. Suppose the H-graph G is (1,1)-win. Let H be a handle of G. The graph $G' = G \setminus H$ is a retract of G with the mapping f defined as follows: $f(X) = f(b) = c$ and $\forall v \in V(G')$, $f(v) = v$. By Theorem 3.2, G' is (1,1)-win. □

Theorem 3.8 *Let G be an H-graph. If there exists a handle H then $G \setminus H$ is an H-graph.*

Proof. Let G be a graph and let H be a handle of G. Suppose $G \setminus H$ is not an H-graph. Then there exists an isometric cycle of length ≥ 5 which has a shortcut through b, X. There are four cases to consider.

(1) The shortcut includes b and at least one vertex from the subgraph X, say x. Also, x and b are immediately preceded by a vertex from $Y \cup \{a\}$ and immediately followed by a vertex from $Y \cup \{c\}$.

Suppose the shortcut includes the path $y_1\ x\ b\ y_2$ where y_1 and y_2 are vertices from Y. The path $y_1\ c\ y_2$ could be used instead. If the shortcut includes the path $a\ x\ b\ y_1$, then the path $a\ c\ y_1$ could be used instead. If the shortcut includes the path $y_1\ x\ b\ c$, the path $y_1\ c$ could be used instead. Finally, if the shortcut includes the path $a\ x\ b\ c$, the path $a\ c$ could be used instead.

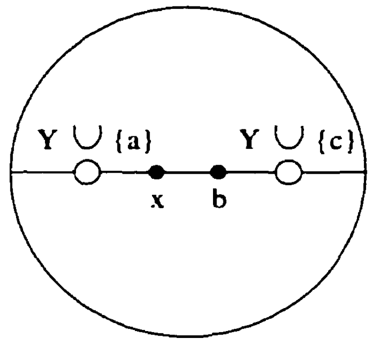

Figure 3.4: Case 1.

(2) The shortcut includes at least one vertex from X but not b. As with the first case. c can be used in the shortcut. The vertices from X are replaced by c in the path.

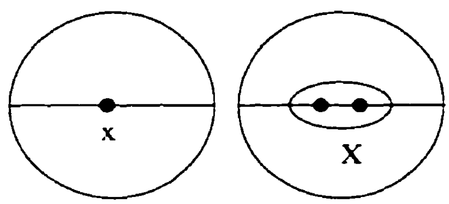

Figure 3.5: Case 2.

(3) The shortcut includes b but no vertices from X. Again. the vertex b can be replaced by c in the shortcut.

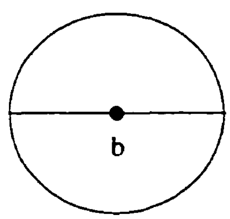

Figure 3.6: Case 3.

Hence all shortcuts through b. X can be rerouted through c. This is a contradiction. Therefore. $G \setminus H$ is an H-graph. □

The previous two theorems tell us that if G is an H-graph with a handle H and G is (1.1)-win. then $G \setminus H$ is an H-graph and $G \setminus H$ is (1.1)-win. Hence the successive removal of handles and corners from a (1.1)-win H-graph G will result in a single vertex.

It was hoped that if such a removal of handles and corners from a (1.1)-win H-graph G resulted in a single vertex. then it could be concluded that G is (1.1)-win. However. this is not the case because a different trap may be needed for every handle in G. This is shown in the next example.

Example: Consider the H-graph G shown in Figure 3.7.

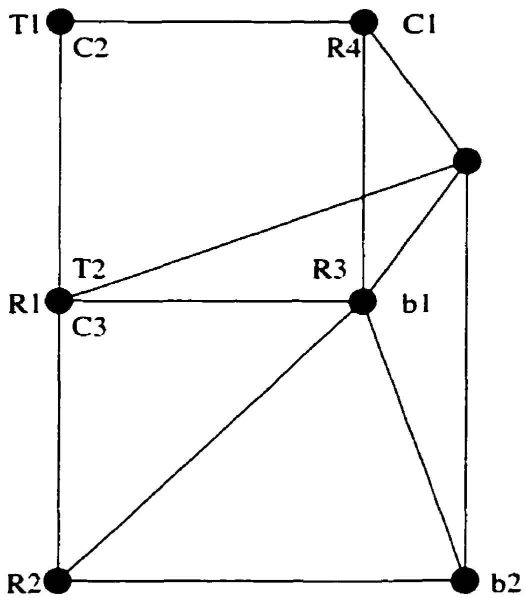

Figure 3.7: A (1.2)-win graph G. The vertex labels indicate the sequence of moves by the cop (C). robber (R) and trap (T).

Suppose the trap is on T_1. the robber is on R_1. and the cop moves onto C_1. Since $R_1 \cup \{b_1\}$ is a handle of $G \setminus \{R_2. b_2\}$. the cop can win on the subgraph $G \setminus \{R_2. b_2\}$ with a single trap by Theorem 3.6. However. a second handle $R_2 \cup \{b_2\}$ is also present in G. Suppose the robber drops down to R_2. There are cycles present on which the robber can move and evade the cop. Therefore. the cop must retrieve the trap if he is to capture the robber on the handle $R_2 \cup \{b_2\}$. The cop moves to T_1 and retrieves the trap. Suppose the robber moves onto R_3. The cop moves to R_1 and drops the trap. The robber is then able to move to R_4 and escape from the handle. Hence one cop and a single trap cannot ensure the capture of a robber on an H-graph G from which handles can be successively removed.

It has been shown that if the successive removal of handles and corners from an H-graph G results in a single vertex . it is not necessarily true that G is (1.1)-win. However. if k handles are removed. a single cop playing on G can win with at most k traps. one for each of the handles.

Theorem 3.9 *Let G be an H-graph. Suppose there is an ordering $S = \{S_1. S_2. \ldots . S_n\}$ such that for each $i < n$. S_i is a handle or a corner in the subgraph induced by the vertices in the set $\{S_i. S_{i+1}. \ldots . S_n\}$. If $|\{S_i : S_i \text{ is a handle}\}| = k$. then one cop playing on G requires at most k traps to win.*

Proof. Let G be an H-graph. It has been shown in Theorem 3.8 that if H is a handle of G. then $G \setminus H$ is also an H-graph. It has been shown in Theorem 3.6 that if the robber is forced to move onto the handle H. the cop can win if he has a trap at his disposal. Therefore once the cop leaves a trap on each of the handles of G. the outcome of the game can be determined by considering the induced subgraph that results from removing these handles. It was shown in Theorem 1.2 that a similar result holds for corners. Inductively. if k handles can be removed from G. as well as a finite number of corners. and the result is a single vertex. then the cop can guarantee a win using at most k traps. one for each handle. □

Now suppose graphs G with isometric cycles of length at least five are considered. The definition of a handle must be expanded to include a subgraph Z as shown in Figure 3.8. This subgraph Z is a copwin graph with the property that there is some copwin ordering $\{z_1. z_2. \ldots . z_n\}$ such that $N_X(z_i) \subseteq N_X(z_j)$ for $j < i$.

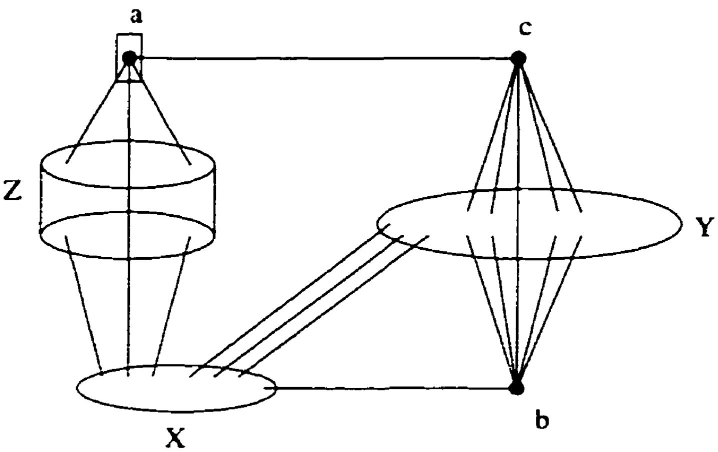

Figure 3.8: A handle $X \cup \{b\}$ under the expanded definition. The trap is indicated by a rectangular box.

Handles $X \cup \{b\}$ defined in this way, together with the subgraphs Y and Z, are copwin graphs. The idea of handles is generalized in the remainder of this chapter. In place of handles, we consider a copwin graph K. The adjacent vertices a and c in the handle definition are outside of K.

Definition 3.3 *Let K be a copwin graph, and let $\{c_1, c_2, \ldots, c_n\}$ be a copwin ordering of K. The vertex c_n will be referred to as the* ***start vertex*** *of this ordering.*

The vertex c_n is called a start vertex because this is the vertex on which the cop begins his winning strategy as described in Section 1.1.5.

Let K be a copwin graph, and let $\mathcal{C} = \{(c_1, c_2, \ldots, c_n) : (c_1, c_2, \ldots, c_n)$ is a copwin ordering of $K\}$. Let $\mathcal{A} \subseteq \mathcal{C}$. Define $A_i(\mathcal{A}) = \{x : x = c_i$ in some copwin ordering in $\mathcal{A}\}$ for $i = 1, 2, \ldots, n$.

Definition 3.4 *Let K be a copwin subgraph of a graph G, and let c and a be adjacent vertices of G outside K. Let $\mathcal{A}$ be the set of copwin orderings of K which end with a start vertex in $N(c)$. Then (K, c, a) is said to be* ***covered*** *if*

(i) for $x \in A_i$ and $y \in A_{i-1}$, $N_{\overline{K}}(y) \subseteq N_{\overline{K}}(x)$,

(ii) $N_{\overline{K}}(A_1, A_2, \ldots, A_n) \subseteq N(a)$ and

(iii) $N_{\overline{K}}(A_1, A_2, \ldots, A_n) \subseteq N(c)$
where $\overline{K} = G \setminus K$.

It is said that the robber is **forced** onto K if the robber is on a vertex $v \in N(c)$ and moves into K when the robber moves onto c.

Theorem 3.10 *Let (K, c, a) be covered. If the robber is forced to move into K then a win for the cop results.*

Proof. Suppose (K, c, a) is covered, and suppose the robber is forced to move into K. If the robber moves onto $x \in K \cap N(c)$, he will be caught on the cop's next move because the cop is on vertex c. Hence, suppose the robber moves into $K \setminus N(c)$.

The cop moves to a and drops the trap. Now the robber cannot move outside K because each vertex x within distance two of c has $N_{\overline{K}}(x) \subseteq N(a)$. This is because (K, c, a) is covered (part(ii)).

The cop returns to c. The robber is on some vertex x. He is unable to move outside K at this time because $d(x, c) \leq 4$ and $N_{\overline{K}}(x) \subseteq N(c)$ by the definition of covered (part (iii)).

The cop moves to a start vertex c_n. It is noted that $c_n \in A_n(\mathcal{A})$. The robber cannot move to a vertex z outside K. This is because the robber is on some vertex x such that $x \in A_i(\mathcal{A})$, $i < n$. By the definition of covered (part (i)), $N_{\overline{K}}(x) \subseteq N_{\overline{K}}(c_n)$. So the robber moves to a vertex in K.

Recursively, the cop uses a copwin ordering to choose a vertex $c_j \in A_j(\mathcal{A})$. The robber is on a vertex $x \in A_k(\mathcal{A})$. Now by Theorem 1.8, the copwin ordering does not allow the robber to use any of the vertices $c_{j+1}, \ldots, c_n$ previously used by the cop. Hence $k < j$. By the definition of covered (part(i)), $N_{\overline{K}}(x) \subseteq N_{\overline{K}}(c_j)$. Hence the robber must move within K. Also by Theorem 1.8, the robber cannot move to any of $c_j, c_{j+1}, \ldots, c_n$.

Hence after a finite number of moves, the robber must move onto a or be apprehended by the cop. Therefore if (K, c, a) is covered and the robber is forced to move into K, a win for the cop results. □

Suppose an alternate definition of the robber being forced into K is used: that is, the robber is forced to remain in K if the robber is in K when the cop moves onto c. The previous theorem holds under this new definition. This is stated as a corollary.

Corollary 3.3 *Let (K, c, a) be covered, and suppose the robber is forced into K in the sense that the robber is in K when the cop moves onto c. Then a win for the cop results.*

As was the case with handles, if the successive removal of k covered, copwin graphs K, as well as a finite number of corners, from a graph G results in a single vertex, then a single cop playing on G can capture the robber using at most k traps.

Chapter 4

Probability Searching Problems

In this chapter, we present several problems formulated in terms of searchers, and sleeping babies and lost dogs. These problems are extensions of the searching game with the exception that it is known there is someone to find. Also, the baby or dog has finite speed. Both the baby or dog and the searchers require one unit of time to move between adjacent vertices. Since the baby or dog is unaware of the efforts of the searchers to find him, he makes no effort to evade the searchers. It is for this reason that these problems are not presented in terms of cops and robbers.

Alternately, we could formulate these problems in terms of a rescue at sea. A rescue boat becomes the searcher, and a missing person takes the place of the lost baby or dog. This naturally would have us consider 2-dimensional grids. However, here we restrict ourselves to paths.

We look at these problems from the perspective of determining a strategy and a place to begin the search that will minimize the expected time required. There is a natural progression of problems. We begin in Section 4.1 by considering a search of a path of length n for a baby who has wandered off and is sleeping somewhere along the path. We assume that it is equally likely that the baby will be found on each of the vertices. It is shown in Corollary 4.1 that the time expected for a search is minimized if the search begins at one of the end vertices of the path.

In Section 4.2, the baby is replaced by a dog in the formulation of the problem.

This is because a more active participant is required. We assume that the dog has been lost and free to wander along the path for some time. This alters the probabilities of the dog being found at each of the vertices. We also assume that the dog falls asleep just before the search begins. This prevents the dog from coming up behind the searcher and moving onto a vertex that has already been searched. Even though the underlying probability distribution has changed. it is shown in Corollary 4.2 that the expected time for a search is still minimized by starting at an end vertex.

Suppose we remove the restriction that the dog falls asleep before the search begins. and therefore. the dog is able to move up behind the searcher. Clearly. the expected time will be minimized if the search begins on an end vertex as this will prevent vertices from having to be searched a second time.

In Section 4.3. an alternate way to prevent the dog from moving up behind the searcher is introduced. We allow the searcher to use a 'trap'. If the dog and the trap occupy the same vertex. the dog is detained and the search is over. Once the trap is laid. the search must proceed on an adjacent vertex.

Two strategies are presented. The first strategy considered has the searcher place the trap and proceed in the direction of the nearest leaf. In Conjecture 4.1. we propose that the expected time to complete a search is minimized when the trap is placed on one of the leaves. Now if the trap is placed on one of the leaves. this is equivalent to beginning the search on a leaf without using the trap. In this case. the trap is of no benefit.

The alternate strategy has the searcher place the trap and proceed in the direction of the farthest leaf. We propose in Conjecture 4.2 that the expected time for a search is minimized using this strategy when the trap is placed on a vertex adjacent to one of the leaves.

These two strategies are compared. and Conjecture 4.3 proposes that the expected time to complete a search is minimized when the alternate strategy is used and the trap is placed on a vertex adjacent to one of the leaves.

In the final section. we change the graph on which the search takes place from a path with n vertices to a cycle with n vertices. As before, the searcher has a single trap to aid in the search. It is proven in Theorem 4.7 that the time expected for a search does not depend on the vertex on which the trap is placed.

4.1 Sleeping Baby Problem

Consider a path P_n with n vertices: that is, $V(P_n) = \{1, 2, \ldots, n\}$ and $E(P_n) = \{(i, i+1) : i = 1, 2, \ldots, n-1\}$. Suppose that a baby has wandered off and is now sleeping on some vertex along the path. We wish to minimize the longest time and the expected time required for a search to locate the baby.

We begin by presenting the strategy that will be used to search for the baby. It will be shown that this strategy is the most efficient method of searching in the sense that the number of edges traversed during a search using this strategy is less than the number required using an alternate strategy.

Strategy. Suppose the search originates at vertex i, $i \in \{1, 2, \ldots, n\}$. The search proceeds in the direction in which the distance to an endpoint is shortest. Once this endpoint has been reached, the search continues in the opposite direction. The vertices are searched in the order they appear.

Proof. To see that this strategy is the most efficient, consider the alternate strategy. Suppose the search originates at vertex i, $i \in \{1, 2, \ldots, n\}$. Suppose the searcher visits n_1 vertices in one direction, then reverses direction and searches n_2 vertices in the other direction, $n_2 > n_1$. Further suppose that the search continues in this way. The searcher visits n_j vertices in one direction, then reverses direction and searches n_{j+1} vertices in the other direction, $n_{j+1} > n_j$. Let the n_j vertices searched during the time between two changes of direction by the searcher be a path p_j. The path p_j has endpoints v_j and v_{j+1}. We note that $v_1 = i$. This is shown in Figure 4.1.

Consider the paths p_{j-1}, p_j, and p_{j+1}. The vertices in the path p_{j-1} are searched three times during this portion of the search. The efficiency of the search is improved by removing the path p_{j-1} and the portion of the path p_j between vertices v_j and

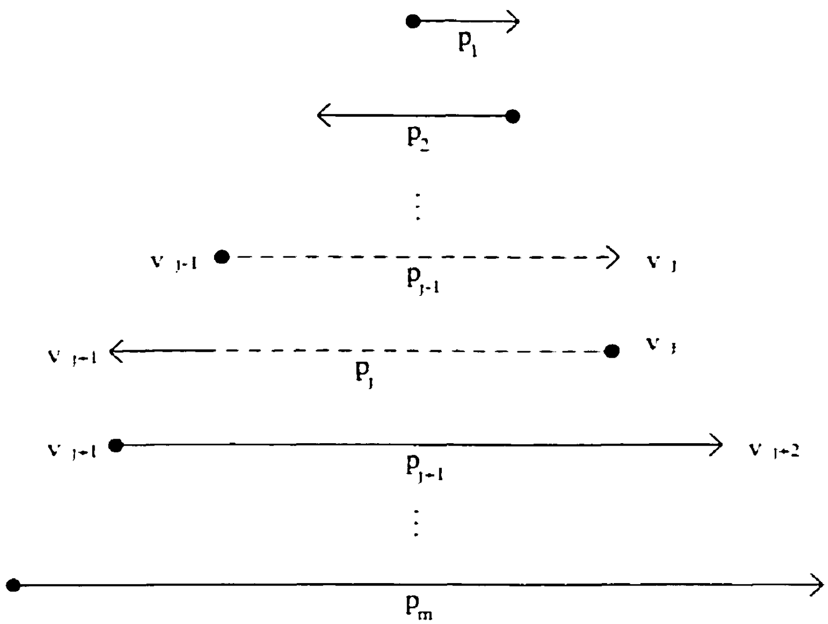

Figure 4.1: Searching strategy.

v_{j-1} (as indicated by dashed lines in Figure 4.1). By induction. after this process of removing inefficient portions of the search is complete. we have a searching strategy in which each edge of the path P_n is covered at most twice. All that remains to be shown is that it is more efficient to search in the direction of the nearer of the two leaves. This is clear since this method of searching minimizes the number of edges that will be covered twice. □

Let's first consider the longest time required for a search. We note that the values along the path are symmetric: that is. the length of the longest path for vertex i is the same as that for vertex $n - i + 1$. The values for the longest paths for each of the vertices are shown in Figure 4.2. Clearly. the longest time is minimized when the search begins at either the first or the last vertex and $n - 1$ units of time are required.

Let's now consider the minimum expected time. We assume that the probability of the baby being located at vertex i is the same for all i. namely $1/n$. Because of symmetry. we need only find the expected times for values of i such that $1 \leq i \leq \frac{n}{2}$

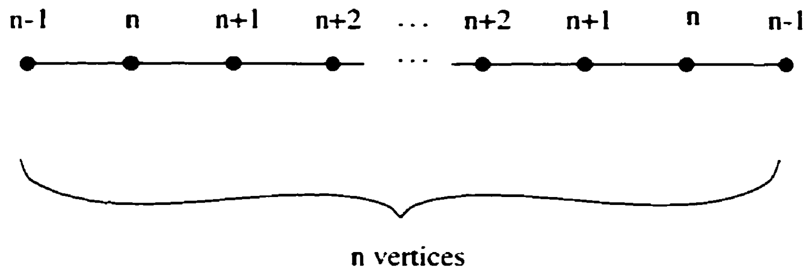

Figure 4.2: Longest times for a search of a path of length n.

for n even or $1 \leq i \leq \frac{n+1}{2}$ for n odd.

Theorem 4.1 *The expected time required to find a baby sleeping somewhere along a path of length n if the search originates at vertex i, $1 \leq i \leq \frac{n}{2}$ for n even and $1 \leq i \leq \frac{n-1}{2}$ for n odd, and continues in the direction of the nearest leaf is $\frac{i(n-i+1)}{n} + \frac{n-3}{2}$.*

Proof. Suppose the search begins at vertex i. The expected value of the time needed for a search is given by $\frac{1}{n}(0) + \frac{1}{n}(1) + \cdots + \frac{1}{n}(i-1) + \frac{1}{n}(2i-1) + \frac{1}{n}(2i) + \frac{1}{n}(2i+1) + \cdots + \frac{1}{n}(n+i-2)$. Clearly the time is 0 if the baby is sleeping at vertex i, 1 if the baby is sleeping at vertex $i-1$, and similarly $i-1$ if the baby is sleeping at vertex 1. Consider now what happens if the baby is situated at vertex $i+1$. The time required for the search is the time to search the vertices between i and 1, return to i and then search $i+1$. The time required is $2i-1$ units. As the other vertices to the right of i are considered, this procedure is repeated adding the necessary units of time that are required since the baby is located further to the right. Finally, if the baby is sleeping at vertex n, $i-1$ units of time are needed to search between vertices i and 1, and then $n-1$ units of time are needed to reach vertex n for a total of $n+i-2$. The expression given above simplifies to $\frac{1}{n}(\sum_{j=1}^{i-1} j + \sum_{j=2i-1}^{n+i-2} j)$ which simplifies to the expression given in the statement of the theorem. $\square$

Corollary 4.1 *The shortest expected time required to find the sleeping baby is $\frac{n-1}{2}$ and occurs when the search begins on either of the two leaves.*

4.2 Lost Dog Problem

Consider a path P_n with n vertices. Suppose we wish to find a dog that is lost somewhere along this path. We wish to determine the time expected to complete this search noting that the dog has been free to wander along the path. We also assume that the dog has fallen asleep just before the search begins.

We must determine the probabilities that the dog will be found at each of the vertices. We first consider the transitional probabilities. If the dog is located at one of the two end vertices, the probability that it stays there is 1/2 and the probability that it moves to the neighboring vertex is 1/2. If the dog is located at any of the other vertices, the probability that it stays there is 1/3 while the probability that it moves to either of the neighboring vertices is 1/3 for each neighbor. All other events have probability 0. Hence the matrix of transitional probabilities is symmetric and appears as shown below.

	a_1	a_2	a_3	a_4	$\dots$	a_{n-3}	a_{n-2}	a_{n-1}	a_n
a_1	1/2	1/2	0	0	$\dots$	0	0	0	0
a_2	1/3	1/3	1/3	0	$\dots$	0	0	0	0
a_3	0	1/3	1/3	1/3	$\dots$	0	0	0	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$	$\vdots$
a_{n-2}	0	0	0	0	$\dots$	1/3	1/3	1/3	0
a_{n-1}	0	0	0	0	$\dots$	0	1/3	1/3	1/3
a_n	0	0	0	0	$\dots$	0	0	1/2	1/2

Now limiting distributions are used to determine the probabilities that the dog will be found at each of the vertices along the path given that the dog has been lost and free to move about for some time. The probabilities are given by the following recursion:

$$\begin{aligned}
(a_1)_{m+1} &= 3(a_1)_m + 2(a_2)_m \\
(a_2)_{m+1} &= 3(a_1)_m + 2(a_2)_m + 2(a_3)_m \\
(a_3)_{m+1} &= 2(a_2)_m + 2(a_3)_m + 2(a_3)_m \\
&\vdots \\
(a_i)_{m+1} &= 2(a_{i-1})_m + 2(a_i)_m + 2(a_{i+1})_m \\
&\vdots \\
(a_p)_{m+1} &= 2(a_{p-1})_m + 2(a_p)_m + 2(a_{p+1})_m
\end{aligned}$$

where $(a_n)_m$ is the mth stage in the recursion for the vertex a_n. $(a_1)_1 = (a_2)_1 = \cdots = (a_p)_1 = 1$. $p = n/2$ if n is even and $p = (n+1)/2$ if n is odd. and $n \geq 4$. The remaining relationships are obtained by symmetry.

Now the actual probabilities are found by solving the equation

$$(a_1, a_2, \ldots, a_n)M = (a_1, a_2, \ldots, a_n)$$

where M is the matrix of transitional probabilities given above. From this equation. we obtain the equalities

$$a_2 = a_3 = \cdots = a_{n-1}$$

and

$$3a_1 + 2a_2 = 6a_1.$$

We also know that

$$2a_1 + (n-2)a_2 = 1.$$

Hence.

$$a_1 = a_n = \frac{2}{3n-2}$$

and

$$a_2 = a_3 = \cdots = a_{n-1} = \frac{3}{3n-2}.$$

This is shown in Figure 4.3.

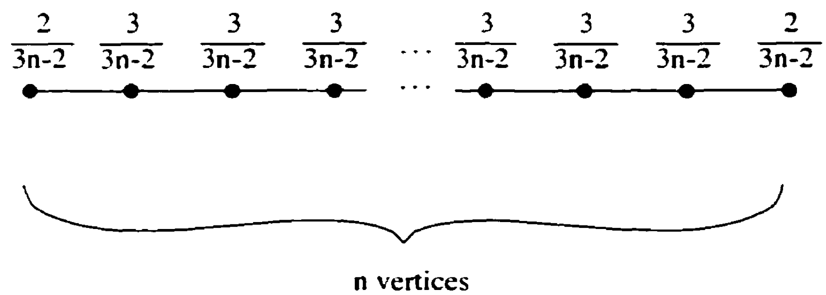

Figure 4.3: Probabilities of finding the lost dog on each of the n vertices.

Now that the distribution of probabilities has been determined, the expected time can be calculated.

Theorem 4.2 *Suppose a dog has been lost and free to roam along a path of length n for some time. Further suppose that if the dog comes to one of the two end vertices, it retains its position with probability 1/2 and moves to the only neighboring vertex with probability 1/2. Similarly, if the dog is located on any of the other $n-2$ vertices, it will retain its position or move to either of the two neighboring vertices each with probability 1/3. Finally suppose that the dog falls asleep just before the search begins. The expected time to find the dog if the search originates at vertex i, $1 \le i \le \frac{n}{2}$ for n even and $1 \le i \le \frac{n+1}{2}$ for n odd, and continues in the direction of the nearest leaf is* $(3i(n-i+\frac{1}{3}) + \frac{n}{2}(3n-11)+3)/(3n-2)$.

Proof. Suppose the search begins at vertex i. The expected time needed for a search is given by $\frac{2}{3n-2}(i-1) + \frac{3}{3n-2}\sum_{j=1}^{i-2} j + \frac{3}{3n-2}\sum_{j=2i-1}^{n+i-3} j + \frac{2}{3n-2}(n+i-2)$. Clearly the probability $\frac{2}{3n-2}$ comes into play only if the dog is located on either the first or the last vertex in the path. The times required for these searches are $i-1$ and $n+i-2$ units of time respectively. All other searches are associated with the probability $\frac{3}{3n-2}$. The sum that includes the values from 1 up to $i-2$ accounts for the vertices to the left of vertex i, except for the first vertex which has already been considered. Similarly, the sum that includes the values from $2i-1$ to $n+i-3$ accounts for the vertices to the right of vertex i with the exception of the nth vertex. This expression simplifies to $\frac{1}{3n-2}(2n+4i-6+3\sum_{j=1}^{i-2} j+3\sum_{j=2i-1}^{n+i-3} j)$ which is equivalent to that given in the statement of the theorem. □

Corollary 4.2 *The shortest expected time needed for such a search is $\frac{n-1}{2}$ and occurs when the search begins at either of the two leaves.*

This is the same conclusion as was drawn in Corollary 4.1 even though the underlying probability distributions are different. Suppose an alternate probability distribution is considered. An interesting question is if the time is minimized when the search begins on a leaf and if this minimum expected time is $\frac{n-1}{2}$.

4.3 Lost Dog Problem with Traps

Consider a path P_n with n vertices. Again we wish to find a dog that has been lost and free to wander along this path for some time. As before we will assume that at any time, it is equally probable that the the dog will remain where it is or move to a neighboring vertex. For the two leaves, this implies that the dog will move with probability 1/2 and remain where it is with probability 1/2. For any of the other vertices, the dog will move to the left or right each with probability 1/3, and will remain where it is with probability 1/3. Hence before the search begins, we assume that the probabilities that the dog is on each of the vertices are given by the limiting distribution found previously and shown in Figure 4.3.

However, we are no longer assuming that the dog falls asleep before the search begins. Hence, the dog is able to move onto vertices that have already been searched. Suppose the search begins at some vertex other than one of the leaves. Immediately after the searcher leaves that vertex, the dog can move up behind the searcher. This is true for all of the vertices visited by the searcher until he reaches one of the leaves. These previously searched vertices have to be searched again. Clearly the searcher can minimize the time expected for a search by beginning at one of the end vertices. This will prevent the dog from moving up behind the searcher, and will eliminate the need to search some vertices more than once.

Another way to prevent the searcher from having to search some vertices more than once is to introduce traps. We allow the searcher to have one trap. Once the trap is placed on a particular vertex, the search continues on the neighboring vertex that is closest to a leaf. We assume, without loss of generality, that the trap is placed on vertex i where $1 \leq i \leq \frac{n}{2}$ if n is even and $1 \leq i \leq \frac{n+1}{2}$ if n is odd. Hence the search begins on vertex $i-1$. Finally, if the dog is caught in the trap, the searcher is immediately aware that the dog has been located and the search is over as the searcher has only to go and pick up the dog.

The first step is to calculate the probabilities that the dog is located at each of the n vertices. These probabilities are updated as the search proceeds.

Let A be the event that the dog is found on vertex k, $1 \leq k \leq n$. Let B be the event that the dog is not found on vertex j, $1 \leq j \leq n-1$, where j is the vertex searched immediately before vertex k. Bayes' Rule tells us

$$P(A|B) = P(B|A)\frac{P(A)}{P(B)}.$$

We notice that $P(B|A) = 1$; that is, we can say with certainty that the dog will not be found on vertex j if it is known that the dog is found on vertex k. Hence,

$$P(A|B) = \frac{P(A)}{P(B)}.$$

We know $P(A_i) = \frac{3}{3n-2}$ where A_i is the event that the dog is found on vertex i, $2 \leq i \leq n-1$. So

$$P(\overline{A}_i) = 1 - \frac{3}{3n-2} = \frac{3n-5}{3n-2}.$$

Also,

$$P(A_{i-1}) = \begin{cases} \frac{3}{3n-2} & , 2 < i \leq n-1 \\ \frac{2}{3n-2} & , i = 2. \end{cases}$$

Hence,

$$P(A_{i-1}|\overline{A}_i) = \begin{cases} \frac{\frac{3}{3n-2}}{\frac{3n-5}{3n-2}} & ,2 < i \leq n-1 \\ \frac{\frac{2}{3n-2}}{\frac{3n-5}{3n-2}} & ,i = 2 \end{cases}$$
$$= \begin{cases} \frac{3}{3n-5} & ,2 < i \leq n-1 \\ \frac{2}{3n-5} & ,i = 2. \end{cases}$$

The other conditional probabilities are calculated similarly.

The first vertices to be searched are those from $i-1$ to 1 inclusive. As the search proceeds down this part of the path, we know that once a vertex has been searched, the probability that the dog will be found at that vertex decreases to 0. This is because the trap prevents the dog from coming up behind the searcher. Hence, the probabilities increase of finding the dog at any of the unsearched vertices. Using Bayes' Rule for conditional probabilities, we obtain the following probabilities where A_k is the event that the dog is found on vertex k, $1 \leq k \leq i-1$.

$$P(A_{i-j}|\overline{A}_{i-j+1}) = \begin{cases} \frac{3}{3n-3j-2} & ,1 \leq j \leq i-2 \\ \frac{2}{3n-3i+1} & ,j = i-1 \end{cases}$$

Now suppose the vertices $i-1$ through 1 have been searched and the dog has not been found and has not been caught in the trap. The searcher knows that the dog must be located on one of the vertices from $i+1$ to n inclusive. All other probabilities are now 0. As was the case when the vertices to the left of the trap were being searched, once a vertex in this part of the path has been searched, the probability of the dog being found on that vertex decreases to 0. Again using Bayes' Rule, the continuously updated probabilities for this part of the path are shown below. They are

$$P(A_{i+1}|\overline{A}_1) = \frac{3}{3n-3i-1}$$

and

$$P(A_{i+j+1}|\overline{A}_{i+j}) = \begin{cases} \frac{3}{3n-3i-3j-1} & ,1 \leq j \leq n-i-2 \\ \frac{2}{3n-3i-3j-1} & ,j = n-i-1 \end{cases}$$

$$= \begin{cases} \frac{3}{3n-3i-3j-1} & .1 \leq j \leq n-i-2 \\ 1 & .j = n-i-1. \end{cases}$$

These probabilities are shown in Figure 4.4 where the trap is represented by a rectangular box.

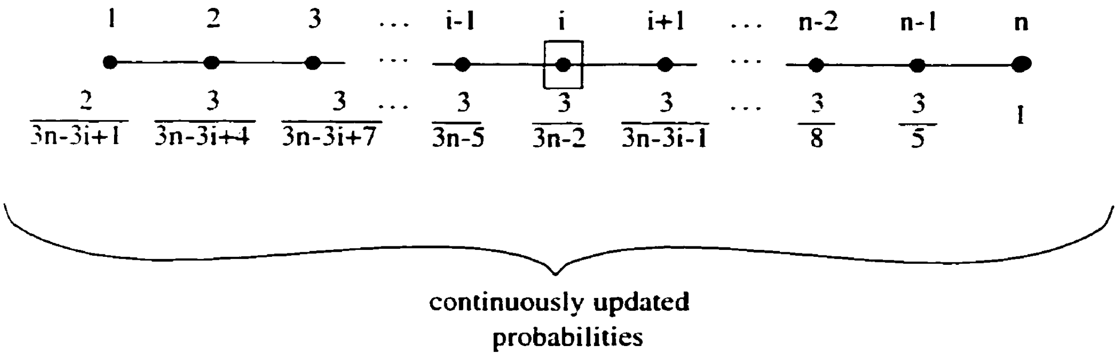

Figure 4.4: Continuously updated probabilities of finding the lost dog on each of the n vertices assuming that the trap is placed on vertex i and the search continues toward the first vertex.

These probabilities are the probabilities of finding the dog on each of the vertices. They are not the probabilities of finding the dog after certain amounts of time have passed. This is because the probabilities that the dog wanders into the trap have not yet been considered.

Suppose j units of time have elapsed where $1 \leq j \leq 2i-2$. At this time. the dog is able to wander into the trap from vertex $i+1$ because no vertices beyond vertex i have yet been searched. The probability of this event is the probability that the dog is located on vertex $i+1$ and moves to the left onto vertex i. namely $\frac{1}{k}\frac{k}{3n-3j-2}$ where $k=3$ if $i \neq n-1$ and $k=2$ otherwise. Hence the probability of the dog wandering into the trap at time j is $\frac{1}{3n-3j-2}$.

Suppose j units of time have elapsed where $j \geq 2i-1$. The search has reached vertex $i+1$. and the probability of the dog wandering into the trap has become 0. This is because all vertices from which the dog could move into the trap have been searched.

Now the probability p_j that the dog is found at time j given that the dog has not previously been found is the sum of the probability that the dog is located on the vertex being searched at this time and the probability that the dog moves onto vertex i. It should be noted that the the probability of the dog being found on the vertex occupied by the searcher at time j where $i \leq j \leq 2i-2$ is 0 since this vertex has already been searched. Also, the probability of the dog moving onto vertex i during the time vertex k is being searched is 0 if $k \geq 2i-1$. The probabilities p_j for all values of j are

$$p_j = \begin{cases} \frac{3}{3n-2} & , j = 0 \\ \frac{4}{3n-3j-2} & , 1 \leq j \leq i-2 \\ \frac{3}{3n-3i+1} & , j = i-1 \\ \frac{1}{3n-3i-1} & , i \leq j \leq 2i-2 \\ \frac{3}{3n+3i-3j-4} & , 2i-1 \leq j \leq n+i-3 \\ 1 & , j = n+i-2. \end{cases}$$

Finally, to determine the probability that the dog is found at time j, the probability that the dog has not been found up to this time must be included. Let l_j denote the probability that the dog will be located at time j. The probability that the dog is located at this time given that it has not been found previously is denoted p_j, and the probability that the dog has not been found prior to this time is denoted a_j. We have $l_j = p_j a_j$. It will be shown that the recursive relationship $l_{j+1} = p_{j+1} a_j (1 - p_j)$ also holds.

Lemma 4.1 *Let l_j be the probability that the dog is found at time j and a_j be the probability that the dog has not been found prior to time j. Given that the dog has not already been found, let p_j be the probability that the dog is found at time j. Then $l_{j+1} = p_{j+1} a_j (1 - p_j)$.*

Proof. Consider the probability l_{j+1} that the dog is found at time $j+1$. It is known that $l_{j+1} = p_{j+1} a_{j+1}$. Let's consider the probability a_{j+1}. This is the probability that the dog has not been found prior to time $j+1$. Alternatively, this can be thought of as the probability that the dog was not found prior to time j and was not found during that time. The probability that the dog was not found prior to time j is a_j. The probability that the dog was not found during time j is $1-p_j$. Hence by the multiplicative rule, the probability that the dog has not been found prior to time $j+1$ can be written $a_j(1-p_j)$; that is, the relationship $l_{j+1} = p_{j+1}a_j(1-p_j)$ holds. □

These probabilities l_j are used to calculate the expected time required for a search using the strategy described in this section. This is the subject of the next theorem which follows directly from the definition of expectation.

Theorem 4.3 *Suppose a dog has been lost and free to wander along a path of length n. Further suppose that a searcher has a single trap to aid in the search. The expected time to find the dog given that the trap is laid on vertex i, $2 \le i \le \frac{n}{2}$ if n is even and $2 \le i \le \frac{n-1}{2}$ if n is odd, and the search proceeds in the direction of the nearest leaf is*

$$\sum_{j=0}^{n+i-2} jp_j(1-p_{j-1})(1-p_{j-2})\cdots(1-p_0)$$

where

$$p_j = \begin{cases} \frac{3}{3n-2} & , j = 0 \\ \frac{4}{3n-3j-2} & , 1 \le j \le i-2 \\ \frac{2}{3n-3i+1} & , j = i-1 \\ \frac{1}{3n-3i-1} & , i \le j \le 2i-2 \\ \frac{3}{3n+3i-3j-4} & , 2i-1 \le j \le n+i-3 \\ 1 & , j = n+i-2. \end{cases}$$

Proof. Suppose that the trap is placed on vertex i where $2 \le i \le \frac{n}{2}$ if n is even and $2 \le i \le \frac{n+1}{2}$ if n is odd. Further suppose that the search proceeds in the direction of the first vertex. The expected time required for such a search is found by multiplying a particular time j by the probability that the dog is found at that time, and summing over all possible values of j to obtain

$$\sum_{j=0}^{n+i-2} jl_j$$

where

$$\begin{aligned} l_j &= a_j p_j. \\ a_j &= a_{j-1}(1 - p_{j-1}). \\ a_0 &= 1. \end{aligned}$$

and

$$p_j = \begin{cases} \frac{3}{3n-2} & . j = 0 \\ \frac{4}{3n-3j-2} & . 1 \le j \le i-2 \\ \frac{3}{3n-3i+1} & . j = i-1 \\ \frac{1}{3n-3i-1} & . i \le j \le 2i-2 \\ \frac{3}{3n+3i-3j-4} & . 2i-1 \le j \le n+i-3 \\ 1 & . j = n+i-2. \end{cases}$$

Now the factor a_j is expanded inductively to obtain $(1-p_{j-1})(1-p_{j-2})\cdots(1-p_0)$. Hence the sum $\sum_{j=0}^{n-i-2} jl_j$ simplifies to the expression given in the statement of the theorem. □

It is desirable for the searcher to know the vertex on which the trap should be placed to minimize the time expected to complete a search for the missing dog. Table 4.1 gives the expected times to search a path P_n. $n \in \{4.5.\ldots 18\}$ for each vertex i on which the trap can be placed. The values given in the table were found using the following code in Maple. The values for n and i have to be entered by the user.

```
> n:= ;
> i:= ;
> p(0):= 3/(3*n-2);
> for j from 1 by 1 to i-2 do p(j):= 4/(3*n-3*j-2) od;
> p(i-1):= 3/(3*n-3*i+1);
```

```
> for j from i by 1 to 2*i-2 do p(j):= 1/(3*n-3*i-1) od;
> for j from 2*i-1 by 1 to n+i-3 do p(j):= 3/(3*n+3*i-3*j-4) od;
> p(n+i-2):=1;
> g := sum('j*p(j)*(product(1-p(k), k=0..j-1))', 'j'=1..n+i-2);
> evalf(g);
```

n	i							
	2	3	4	5	6	7	8	9
4	1.5480							
5	2.1911	1.9418						
6	2.7805	2.6785						
7	3.3416	3.3502	3.1036					
8	3.8860	3.9786	3.8314					
9	4.4197	4.5778	4.5138	4.2676				
10	4.9462	5.1565	5.1618	4.9907				
11	5.4676	5.7205	5.7837	5.6789	5.4326			
12	5.9851	6.2735	6.3857	6.3387	6.1528			
13	6.4999	6.8181	6.9721	6.9756	6.8446	6.5982		
14	7.0124	7.3562	7.5462	7.5940	7.5123	7.3164		
15	7.5232	7.8890	8.1105	8.1972	8.1598	8.0106	7.7640	
16	8.0325	8.4176	8.6667	8.7879	8.7903	8.6840	8.4808	
17	8.5407	8.9428	9.2162	9.3682	9.4064	9.3393	9.1768	8.9301
18	9.0480	9.4650	9.7602	9.9398	10.0103	9.9791	9.8545	9.6458

Table 4.1: Expected times for a search of P_n for each possible value of i. the vertex on which the trap is placed.

We conclude this section by considering the case not included in the previous analysis. Suppose the trap is placed on the first vertex rather than the ith vertex. The probabilities that the dog is located on each of the n vertices are shown in Figure 4.5.

In this case. it is impossible for the dog to wander into the trap. Hence the probabilities p_j are as shown in Figure 4.5. The probability l_j of finding the dog at a particular time j is found using the recursive relationship $l_j = p_j a_j$ where $a_j = a_{j-1}(1 - p_{j-1})$. These probabilities l_j are used to determine the expected time.

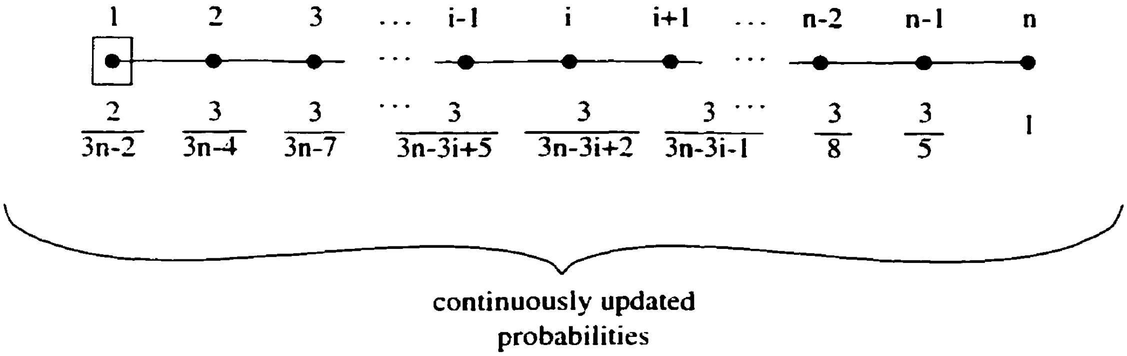

Figure 4.5: Continuously updated probabilities of finding the lost dog on each of the n vertices when the trap is placed on the first vertex.

Theorem 4.4 *Suppose a dog has been lost and free to wander along a path of length n. Further suppose that a searcher has a single trap to aid in the search and that the trap is laid on the first vertex. The expected time to find the dog is $\frac{n-1}{2}$.*

Proof. Suppose that the trap is placed on the first vertex and the search begins on the second vertex. The expected time is found by multiplying a particular time j by the probability that the dog is found at that time, and summing over all possible values of j to obtain $\sum_{j=0}^{n-1} jl_j$ where $l_j = a_j p_j$, $a_j = a_{j-1}(1 - p_{j-1})$ and

$$p_j = \begin{cases} \frac{2}{3n-2} & , j = 0 \\ \frac{3}{3n-3j-1} & , 1 \leq j \leq n-2 \\ 1 & , j = n-1. \end{cases}$$

Now the factor a_j is expanded inductively to obtain $(1-p_{j-1})(1-p_{j-2})\cdots(1-p_1)(1-p_0)$, and so this sum simplifies to

$$\sum_{j=0}^{n-1} jp_j(1 - p_{j-1})(1 - p_{j-2})\cdots(1 - p_1)(1 - p_0).$$

Now consider the first $n-2$ terms in this sum. Each such term can be expanded to obtain

$$j(\frac{3}{3n-3j-1})(1-\frac{3}{3n-3j+2})(1-\frac{3}{3n-3j+5})\cdots(1-\frac{3}{3n-4})(1-\frac{2}{3n-2})$$

which can be simplied to obtain

$$j(\frac{3}{3n-3j-1})(\frac{3n-3j-1}{3n-3j+2})(\frac{3n-3j+2}{3n-3j+5})\cdots(\frac{3n-7}{3n-4})(\frac{3n-4}{3n-2})$$

or equivalently

$$\frac{3j}{3n-2}.$$

Hence.

$$\sum_{j=0}^{n-2} jl_j = \frac{3}{3n-2}\sum_{j=0}^{n-2} j$$

which can be simplied to obtain

$$\frac{3n^2-9n+6}{2(3n-2)}.$$

Similarly. the last term in the sum ($j = n-1$) can be expanded to obtain

$$(n-1)(1-\frac{3}{5})(1-\frac{3}{8})\cdots(1-\frac{3}{3n-7})(1-\frac{3}{3n-4})(1-\frac{2}{3n-2}).$$

This expression can be simplied to

$$(n-1)(\frac{2}{5})(\frac{5}{8})\cdots(\frac{3n-10}{3n-7})(\frac{3n-7}{3n-4})(\frac{3n-4}{3n-2})$$

or equivalently

$$\frac{2(n-1)}{3n-2}.$$

Hence.

$$\begin{aligned}\sum_{j=0}^{n-1} jl_j &= \frac{3n^2-9n+6}{2(3n-2)} + \frac{2(n-1)}{3n-2} \\ &= \frac{3n^2-5n+2}{2(3n-2)} \\ &= \frac{(3n-2)(n-1)}{2(3n-2)} \\ &= \frac{n-1}{2}.\end{aligned}$$

This is the expression given in the statement of the theorem. □

It appears that when using the strategy described in this section. the expected time for a search is minimized when the trap is placed on the first vertex for $n \neq 5$. This is stated as a conjecture. and is tested for some large values of n. The results are shown in Table 4.2.

Conjecture 4.1 *Suppose a dog has been lost and free to wander along a path of length n. Further suppose that a searcher has a single trap to aid in the search. and that the search proceeds in the direction of the nearest leaf. The expected time to find the dog given that the trap is laid on vertex i. $1 \leq i \leq \frac{n}{2}$ if n is even and $1 \leq i \leq \frac{n-1}{2}$ if n is odd. is minimized when $i = 1$ for $n \neq 5$.*

It is known that Conjecture 4.1 is true for values of $n \leq 250$.

i	n				
	50	100	150	200	250
1	24.5000	49.5000	74.5000	99.5000	124.5000
2	25.1249	50.1460	75.1529	100.1564	125.1584
$\lceil \frac{n}{2} \rceil$	28.3049	57.4708	86.6382	115.8060	144.9739

Table 4.2: Comparison of expected times for $i = 1, 2$, and $\lceil \frac{n}{2} \rceil$.

Consider the strategy presented here when the trap is placed on the first vertex. It is equivalent to the search beginning on the first vertex when no trap is available. Hence if $n \neq 5$. it is not to the searcher's benefit to use a trap.

4.4 Lost Dog Problem with Traps - The Alternate Strategy

In this section. an alternate strategy that can be used by a searcher with a single trap is introduced. Rather than placing the trap on some vertex i and then proceeding to search in the direction of the nearest of the two leaves. the searcher places the trap on the second vertex and proceeds to search in the direction of the farthest leaf. This is because the probability that the dog is on the first vertex is small in comparison to the probability that it is on some vertex to the right of the trap. and if the dog is located on the first vertex. the probability is one half that the dog will wander into the trap rather than remain on the first vertex. Hence. the probability that the searcher will have to retrace his steps near the end of the search and search the first vertex is small. The expected time for a search using this strategy will be compared with the time expected using the strategies presented in the previous section.

The probabilities that the dog is located on each of the n vertices are shown in Figure 4.6. They are calculated using Bayes' Rule as in Section 4.3.

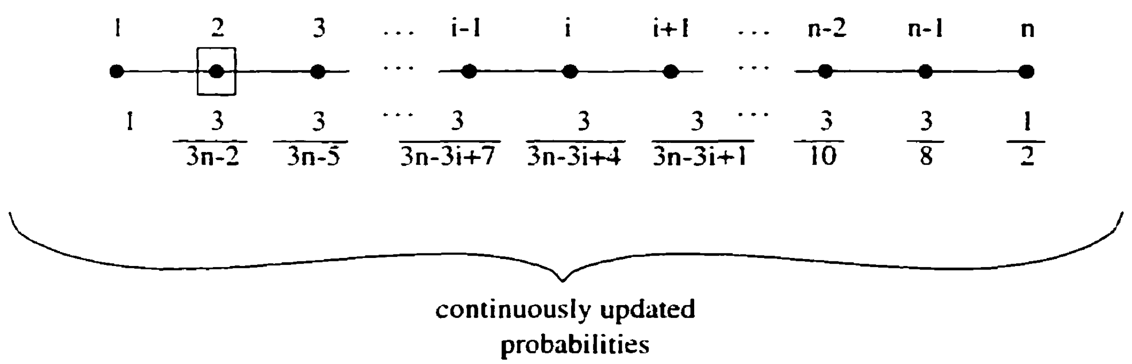

Figure 4.6: Continuously updated probabilities of finding the lost dog on each of the n vertices if the trap is placed on the second vertex and the alternate strategy is being used.

Now consider the probability p_j that the dog is found at a particular time j given that the dog had not been found prior to that time. This probability is found by adding the probability that the dog is found on the vertex being searched at that time to the probability that the dog wanders into the trap. Consider the probability that the dog wanders into the trap. Now at time j, the probability that the dog is located on the first vertex is $\frac{2}{3n-3j-2}$ and the probability that the dog moves to the right is 1/2. Hence the probability of the dog moving into the trap is $\frac{1}{3n-3j-2}$.

It should be noted that during the times from 1 to $n-2$ inclusive, given that the dog is still missing, the probability of the dog being found includes contributions from both the probability that the dog is on the vertex being searched at this time and the probability that the dog wanders into the trap. During the times from $n-1$ to $2n-4$, the probability that the dog is located on the vertex being searched is 0 since the vertices being searched during these times have already been searched. Hence the probability p_j depends solely on the probability that the dog wanders into the trap. At time $2n-3$, given that the dog has not been found, the probability of the dog being on the first vertex is 1. These probabilities p_j are

$$p_j = \begin{cases} \frac{3}{3n-2} & , j = 0 \\ \frac{4}{3n-3j-2} & , 1 \leq j \leq n-3 \\ \frac{3}{4} & , j = n-2 \\ \frac{1}{2} & , n-1 \leq j \leq 2n-4 \\ 1 & , j = 2n-3. \end{cases}$$

Hence the probability l_j of finding the dog at a particular time j is found using the recursive relationship $l_j = p_j a_j$ where $a_j = a_{j-1}(1-p_{j-1})$. These probabilities l_j are used to determine the expected time.

Theorem 4.5 *Suppose a dog has been lost and free to wander along a path of length n. Further suppose that a searcher has a single trap to aid in the search. The expected time to find the dog given that the trap is laid on the second vertex and the search proceeds in the direction of the nth vertex is*

$$\sum_{j=0}^{2n-3} j p_j (1-p_{j-1})(1-p_{j-2}) \cdots (1-p_1)(1-p_0)$$

where

$$p_j = \begin{cases} \frac{3}{3n-2} & , j = 0 \\ \frac{4}{3n-3j-2} & , 1 \le j \le n-3 \\ \frac{3}{4} & , j = n-2 \\ \frac{1}{2} & , n-1 \le j \le 2n-4 \\ 1 & , j = 2n-3. \end{cases}$$

Proof. Suppose that the trap is placed on the second vertex and the search continues on the third. The expected time is found by multiplying a particular time j by the probability that the dog is found at that time, and summing over all possible values of j to obtain $\sum_{j=0}^{2n-3} jl_j$ where $l_j = a_j p_j$ and $a_j = a_{j-1}(1 - p_{j-1})$. The factor a_j is expanded inductively as before to obtain $(1-p_{j-1})(1-p_{j-2})\cdots(1-p_1)(1-p_0)$, and so this sum simplies to the one given in the statement of the theorem. □

Maple was used to compute the expected times for a search of the path P_n for various values of n using the strategy presented in this section. These values are given in Table 4.3. The Maple code used to obtain these values is included.

```
> n:= ;
> p(0):=3/(3n-2);
> for j from 1 by 1 to n-3 do p(j):= 4/(3*n-3*j-2) od;
> p(n-2) := 3/4;
> for j from n-1 by 1 to 2*n-4 do p(j):=1/2 od;
> p(2*n-3):= 1;
> g := sum('j*p(j)*(product(1-p(k), k=0..j-1))', 'j'=1..2*n-3);
> evalf(g);
```

We conclude this section by considering a more general form of the strategy presented here. Suppose that instead of placing the trap on the second vertex and proceeding in the direction of the nth vertex, the searcher places the trap on the ith

n	Expected Time
4	1.1313
5	1.5213
6	1.9272
7	2.3411
8	2.7594
9	3.1807
10	3.6037
11	4.0280
12	4.4532
13	4.8791
14	5.3054
15	5.7321
16	6.1591
17	6.5863
18	7.0137
19	7.4413
20	7.8690
21	8.2968
22	8.7246
23	9.1526
24	9.5806
25	10.0086

Table 4.3: Expected times for a search using the alternate strategy.

vertex and proceeds in the direction of the nth vertex, where $3 \leq i \leq \frac{n}{2}$ if n is even and $3 \leq i \leq \frac{n-1}{2}$ if n is odd.

Given that the dog has not been found previously, the probabilities that the dog is located on each of the n vertices are shown in Figure 4.7. They are calculated using Bayes' Rule as in Section 4.3.

The probability p_j that the dog is found at a particular time j given that the dog has not been found prior to that time is found by adding the probability that the dog is on the vertex being searched at that time to that probability that the dog wanders into the trap. The probability that the dog wanders in the trap at time j is calculated as before. If $1 \leq j \leq n-i$, the probability of the dog moving into the trap is $\frac{1}{3n-3j-2}$.

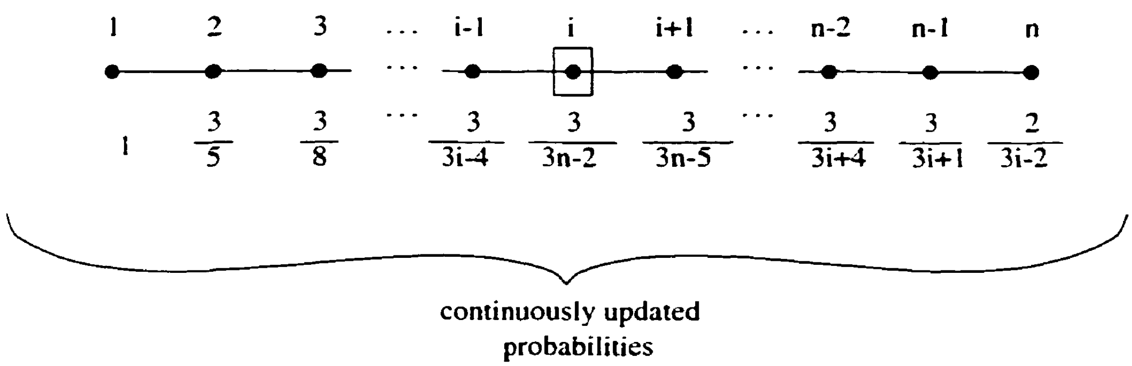

Figure 4.7: Continuously updated probabilities of finding the lost dog on each of the n vertices if the trap is placed on vertex i and the alternate strategy is being used.

If $n - i + 1 \leq j \leq 2n - 2i$, this probability is $\frac{1}{3i-4}$. Otherwise, the probability is 0 since the vertices adjacent to the trap have been searched. The probabilities p_j are

$$p_j = \begin{cases} \frac{3}{3n-2} & , j = 0 \\ \frac{4}{3n-3j-2} & , 1 \leq j \leq n - i - 1 \\ \frac{3}{3i-2} & , j = n - i \\ \frac{1}{3i-4} & , n - i + 1 \leq j \leq 2n - 2i \\ \frac{3}{6n-3i-3j-1} & , 2n - 2i + 1 \leq j \leq 2n - i - 2 \\ 1 & , j = 2n - i - 1. \end{cases}$$

Hence the probability l_j of finding the dog at a particular time j is found using the recursive relationship $l_j = p_j a_j$ where $a_j = a_{j-1}(1 - p_{j-1})$. These probabilities l_j are used to determine the expected time.

Theorem 4.6 *Suppose a dog has been lost and free to wander along a path of length n. Further suppose that a searcher has a single trap to aid in the search. The expected time to find the dog given that the trap is laid on vertex i, $3 \leq i \leq \frac{n}{2}$ if n is even and $3 \leq i \leq \frac{n-1}{2}$ if n is odd, and the search proceeds in the direction of the nth vertex is*

$$\sum_{j=0}^{2n-i-1} j p_j (1 - p_{j-1})(1 - p_{j-2}) \cdots (1 - p_1)(1 - p_0)$$

where

$$p_j = \begin{cases} \frac{3}{3n-2} & , j = 0 \\ \frac{4}{3n-3j-2} & , 1 \le j \le n-i-1 \\ \frac{3}{3i-2} & , j = n-i \\ \frac{1}{3i-4} & , n-i+1 \le j \le 2n-2i \\ \frac{3}{6n-3i-3j-1} & , 2n-2i+1 \le j \le 2n-i-2 \\ 1 & , j = 2n-i-1. \end{cases}$$

Proof. Suppose that the trap is placed on vertex i, $3 \le i \le \frac{n}{2}$ if n is even and $3 \le i \le \frac{n-1}{2}$ if n is odd, and the search continues on vertex $i+1$. The expected time is found by multiplying a particular time j by the probability that the dog is found at that time, and summing over all possible values of j to obtain $\sum_{j=0}^{2n-i-1} jl_j$ where $l_j = a_j p_j$ and $a_j = a_{j-1}(1-p_{j-1})$. The factor a_j is expanded inductively as before to obtain $(1-p_{j-1})(1-p_{j-2})\cdots(1-p_1)(1-p_0)$, and so this sum simplies to the one given in the statement of the theorem. □

Maple was used to compute the expected times for a search of the path P_n for various values of n and i using the strategy presented above. These values are given in Table 4.4. The Maple code is included.

```
> n:= ;
> i:= ;
> p(0):= 3/(3*n-2);
> for j from 1 by 1 to n-i-1 do p(j):= 4/(3*n-3*j-2) od;
> p(n-i):= 3/(3*i-2);
> for j from n-i+1 by 1 to 2*n-2*i do p(j):= 1/(3*i-4) od;
> for j from 2*n-2*i+1 by 1 to 2*n-i-2 do p(j):= 3/(6*n-3*i-3*j-1) od;
> p(2*n-i-1):=1;
> g := sum('j*p(j)*(product(1-p(k), k=0..j-1))', 'j'=1..2*n-i-1);
> evalf(g);
```

n	i							
	3	4	5	6	7	8	9	10
6	2.3213							
7	2.7018							
8	3.0870	3.5036						
9	3.4774	3.8936						
10	3.8728	4.2804	4.6799					
11	4.2727	4.6672	5.0793					
12	4.6762	5.0556	5.4719	5.8530				
13	5.0830	5.4461	5.8613	6.2600				
14	5.4926	5.8392	6.2496	6.6587	7.0243			
15	5.9044	6.2347	6.6379	7.0524	7.4373			
16	6.3183	6.6328	7.0271	7.4431	7.8415	8.1944		
17	6.7339	7.0332	7.4177	7.8325	8.2397	8.6123		
18	7.1509	7.4357	7.8098	8.2213	8.6338	9.0213	9.3677	
19	7.5692	7.8403	8.2035	8.6103	9.0255	9.4237	9.7857	
20	7.9886	8.2467	8.5990	8.9999	9.4156	9.8214	10.1987	10.5326

Table 4.4: Expected times for a search when the trap is laid on vertex i and the search proceeds on vertex $i+1$.

It appears that when using the strategy described in this section, the expected time for a search is minimized when the trap is laid on the second vertex. This is stated as a conjecture. Table 4.5 compares the expected times when $i = 2$, 3, and $\lfloor \frac{n}{2} \rfloor$ for some large values of n.

i	n				
	50	100	150	200	250
2	20.7176	42.1442	63.5722	85.0005	106.4290
3	20.7520	42.1576	63.5800	85.0058	106.4329
$\lfloor \frac{n}{2} \rfloor$	28.0447	57.2165	86.3859	115.5547	144.7232

Table 4.5: Comparison of expected times using the alternate strategy.

Conjecture 4.2 *Suppose a dog has been lost and free to wander along a path of length n. Further suppose that a searcher has a single trap to aid in the search. The* *expected time to find the dog given that the trap is laid on vertex i.* $2 \leq i \leq \frac{n}{2}$ *if n is even and* $2 \leq i \leq \frac{n-1}{2}$ *if n is odd. and the search proceeds on vertex* $i+1$ *is minimized when* $i = 2$.

It is known that Conjecture 4.2 is true for values of $n \leq 250$.

4.5 Lost Dog Problem with Traps: Most Efficient Strategy

The previous sections have presented two strategies for searching for a lost dog on a path P_n of length n when the dog has been lost and free to wander for some time. and a trap is available to aid in the search. Conjectures have been made about the vertex on which the trap should be placed to minimize the expected time for a search using each of these strategies.

If the trap is laid on vertex i. $1 \leq i \leq \frac{n}{2}$ if n is even and $1 \leq i \leq \frac{n+1}{2}$ if n is odd. and the search proceeds toward the first vertex. it has been conjectured that the expected time is minimized when $i = 1$. If the trap is laid on vertex i. $2 \leq i \leq \frac{n}{2}$ if n is even and $1 \leq i \leq \frac{n-1}{2}$ if n is odd. and the search proceeds toward the nth vertex. it has been conjectured that the expected time is minimized when $i = 2$.

In this section. these two strategies are compared for large values of n to determine the strategy which minimizes the time expected to complete a search for the missing dog. The results are shown in Table 4.6.

i	n					
	50	100	150	200	250	500
1	24.5000	49.5000	74.5000	99.5000	124.5000	249.5000
2	20.7176	42.1442	63.5722	85.0005	106.4290	213.5716

Table 4.6: Comparison of the expected times when using the two most efficient strategies.

It appears that it is more efficient to place the trap on the second vertex and proceed in the direction of the nth vertex. This is stated as a conjecture.

Conjecture 4.3 *Suppose a dog has been lost and free to wander along a path of length n. Further suppose that a searcher has a single trap to aid in the search. The expected time for the search is minimized if the searcher places the trap on the second vertex and then proceeds to search in the direction of the nth vertex.*

This conjecture is known to be true for values of $n \leq 500$.

4.6 Lost Dog Problem on a Cycle

Consider a cycle C_n with n vertices. Suppose a dog has been lost and free to wander on C_n for some time. We assume that the probabilities of the dog moving to the left or right or staying still are equal. Suppose that the searcher has one trap to aid in the search and the trap is laid on vertex i. Now the search must begin on one of the vertices adjacent to vertex i. Without loss of generality, assume that the search begins on vertex $i + 1$. As before, the probabilities are continuously updated as the search continues.

The first step is to determine the probabilities that the dog is on each of the vertices before the search starts. Clearly, these probabilities are all equal since the dog is equally likely to move to the left or right or to stay still. Hence, for each of the vertices there is a $1/n$ chance that the dog will be found there.

Now consider what happens once the trap is laid. If the dog is located on the ith vertex, it will be found when the searcher goes to that vertex to lay the trap. If the dog is not found there, the searcher uses that information to update the probabilities. The probability that the dog is located on the ith vertex becomes 0 and the remaining probabilities become $1/(n-1)$. This is intuitive and can be easily verified using Bayes' Rule for conditional probabilities.

Similarly, as the search continues the probabilities that the dog is located on any of the previously searched vertices become 0, and the probabilities that the dog is located on any of the other vertices increase. These continuously updated probabilities are represented by the formulae which follow where A_j is the event that the dog is found on vertex j. $j \in \{1, 2, \ldots, n\}$.

$$P(A_j) = \begin{cases} \frac{1}{n+i-j} & , j = i+1, \ldots, n \\ \frac{1}{i-j} & , j = 1, \ldots, i-1 \end{cases}$$

These probabilities are shown in Figure 4.8.

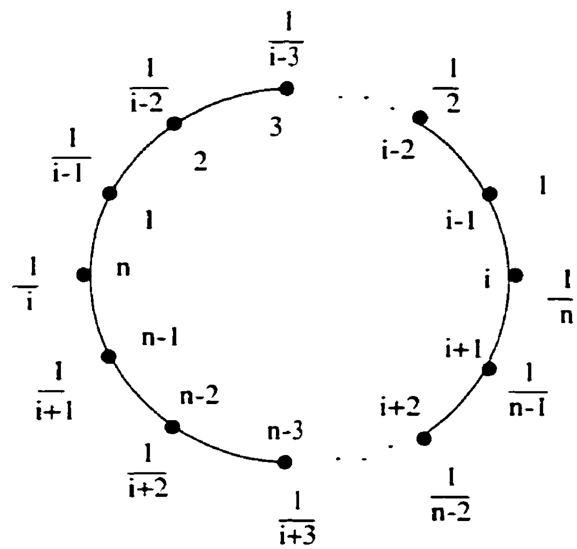

Figure 4.8: Continuously updated probabilities of finding the lost dog on each of the n vertices of the cycle given that the dog has not been found prior to the time these vertices are searched.

It should be noted here that these continuously updated probabilities are not the probabilities that the dog will be found while a search of these vertices is proceeding. This is because the probability that the dog wanders into the trap has not been included.

Suppose j units of time have passed. The probability that the dog wanders into the trap is $\frac{1}{n-j}$. the probability that the dog is on vertex $i-1$. multiplied by 1/3. the probability that the dog moves onto vertex i. Hence the probability of this event is $\frac{1}{3(n-j)}$.

Now p_j is the probability that the dog is found after j units of time given that it has not been found previously. These probabilities p_j are

$$p_j = \begin{cases} \frac{4}{3(n-j)} & , 1 \leq j \leq n-2 \\ 1 & , j = n-1. \end{cases}$$

Let l_j be the probability that the dog is found after j units of time. As shown previously, $l_j = p_j a_j$ where $a_j = a_{j-1}(1 - p_{j-1})$. These are the probabilities used to determine the expected time required for such a search.

Theorem 4.7 *Suppose a dog has been lost and free to wander along a cycle of length n, and that the dog is equally likely to move to the right or left or to retain its position. Further suppose that the searcher has a single trap, that the search continues on a vertex adjacent to the one on which the trap is placed, and that the search ends if the dog is found by the searcher or becomes caught by the trap. The expected time required for a search to find the dog does not depend on the location of the trap and is given by*

$$\sum_{j=0}^{n-1} j p_j (1 - p_{j-1})(1 - p_{j-2}) \cdots (1 - p_1)(1 - p_0)$$

where

$$p_j = \begin{cases} \frac{4}{3(n-j)} & , 1 \leq j \leq n-2 \\ 1 & , j = n-1. \end{cases}$$

Proof. Assume that the trap is placed on vertex i and the search continues on vertex $i+1$. The expected time is found by multiplying a particular time j by the probability that the dog is found at that time, and summing over all values of j to obtain $\sum_{j=0}^{n-1} j l_j$ where $l_j = a_j p_j$ and $a_j = a_{j-1}(1 - p_{j-1})$. The factor a_j is expanded as before to obtain $(1 - p_{j-1})(1 - p_{j-2}) \cdots (1 - p_1)(1 - p_0)$, and so this sum simplifies to the expression given in the statement of the theorem. The given expression clearly does not depend on i, the vertex on which the trap was placed. □

Bibliography

[1] Aigner M. and Fromme M., *A Game of Cops and Robbers.* Discrete Appl. Math. **8** (1984), 1-12.

[2] Anstee R. P. and Farber M., *On Bridged Graphs and Cop-win Graphs.* J. Combin. Theory (Ser. B) **44** (1988), 22-28.

[3] Berarducci A. and Intrigila B., *On the Cop Number of a graph.* Advances in Appl. Math. **14** (1993), 389-403.

[4] Chepoi, Victor, *Bridged Graphs are Cop-win Graphs: An Algorithmic Proof.* J. Combin. Theory (Ser. B) **69** (1997), 97-100.

[5] Ellis J. A., Sudborough, I. H. and Turner J. S., *The Vertex Separation and Search Number of a Graph.* Inform. and Conput. **113** (1994), 50-79.

[6] Fitzpatrick, Shannon L., Ph.D. Thesis, Dalhousie University, 1997.

[7] Hartnell B. L., Rall D. F. and Whitehead C. A., *The Watchman's Walk Problem: An Introduction.* Congressus Numerantium **130** (1998), 149-155.

[8] Neufeld S., M.Sc. Thesis, Dalhousie University, 1991.

[9] Neufeld S. and Nowakowski R., *A Game of Cops and Robbers Played on Products of Graphs.* Discrete Math. **186** (1998), 253-268.

[10] Nowakowski R. and Winkler P., *Vertex to Vertex Pursuit in a Graph.* Discrete Math. **43** (1983), 23-29.

[11] Parsons T. D., *Pursuit Evasion in a Graph.* Lecture Notes in Math. **642** (Springer), Berlin 1978.

[12] Quilliot A., *Thèse d'Etat.* Université de Paris VI, 1983.

[13] West, Douglas B., *Introduction to Graph Theory.* Prentice-Hall Inc., NJ, 1996.