WHY THE BAD GUYS ALWAYS GET CAUGHT

by

Justin James Boutilier

Thesis

submitted in partial fulfillment of

the requirements of the Degree of

Bachelor of Science with Honours in

Mathematics and Statistics.

Acadia University

May 2013

This thesis by Justin Boutilier

is accepted in its present form by the

Department of Mathematics and Statistics

as satisfying the thesis requirements for the Degree of

Bachelor of Science with Honours.

Approved by the Thesis Supervisor

Dr. N. Clarke   Date

Approved by the Head of the Department

Dr. J. Hooper   Date

Approved by the Honours Committee

Dr. P. Ranjan   Date

I, Justin Boutilier, hereby grant permission to the University Librarian at Acadia University to provide copies of the thesis, on request, on a non-profit basis.

Signature of Author

Signature of Supervisor

Date

Acknowledgements

First and foremost, I would like to thank Dr. Nancy Clarke for her continued support, encouragement, and patience. I would like to thank her for giving me this opportunity, for believing in me academically, and for making this thesis possible.

I would like to thank Jeff Hooper for his comments and suggestions. I would also like to thank Acadia University and the Department of Mathematics and Statistics for providing me with funding, and the opportunity to learn and mature as a student.

Finally, on a more personal note, I would like to thank Neil Spencer, Julia Whidden and Nick DeAdder, for all of their help and support throughout the writing and research of this thesis. Thank you.

Table of Contents

List of Figures

List of Tables

Abstract

The Cops and Robber game is a discrete time vertex-to-vertex pursuit game played on some graph, G. The original game is played with one cop, one robber and perfect information. This thesis examines a modified version of the Cops and Robber game with partial information and k cops. The information is provided by two types of devices, those placed on vertices and those placed on edges. The information devices can either provide the direction of the robber or not, allowing us to partition the problem into four cases. The main focus of this thesis is to fix the amount of information provided and determine the number of cops required to apprehend the robber. We will focus primarily on a specific class of graphs, grids. A new strategy for placing the information devices is developed and bounds on the number of cops are given for each of the four cases. In the fourth section, we look at the problem from the opposite perspective; we fix the number of cops at one and determine the information required for the robber to be apprehended. A new strategy is developed which improves the bounds on the amount of information needed for full complete k-ary trees with one cop.

Chapter 1

Introduction

The original game of Cops and Robber was first introduced independently by Nowakow--ski and Winkler [8], and Quilliot [9] in 1983. The original game consists of one cop and one robber playing a discrete vertex-to-vertex pursuit game on a finite reflexive graph, G. The cop begins by choosing a starting vertex, after which the robber chooses his starting vertex. The cop and the robber may only be located on vertices and take turns moving from vertex to vertex along edges. However, since G is reflexive, meaning that there is a loop at every vertex, this allows the players to traverse the loop and stay at the same vertex (called 'passing'). The original game is played with perfect information, meaning that both players can see one another's moves and the entire graph. The goal of the game is for the cop to capture the robber and this occurs when the cop and robber simultaneously occupy the same vertex after a finite number of moves.

In this thesis, a modified version of the original game, introduced in [8] and [9], will be examined where the cop will no longer be playing the game with perfect information. Instead, we will have information-providing devices that are placed on edges or vertices. It should be noted that the robber will continue to play with perfect information. Now, there are four of these devices: cameras, alarms, photo radar units with direction and photo radar units without direction. **Cameras** are placed on vertices and give two types of information, location and direction. This means that when a robber lands on a vertex with a camera, the cop is aware he is occupying said vertex and the cop knows the direction in which the robber is moving once he has left the vertex. In other words, he knows along which edge the robber has left the camera vertex and so he knows the next vertex to which the robber moves. **Alarms** are also placed on vertices and only give the location of the robber. If a robber lands on a vertex with an alarm, the cop is aware that he is occupying that vertex but won't know the direction the robber has gone, only that he has left. A **photo radar unit** is a detection device that is placed on an edge instead of a vertex. A **photo radar unit without direction** will only inform the cop that the robber has traversed the edge. However, a **photo radar unit with direction** will inform the cop that the robber has traversed the edge and in which direction he is going. This means that the cop will know the next vertex that the robber plans to occupy.

The main focus of this thesis is to fix the amount of information provided and determine the number of cops required to apprehend the robber. This will be done in general on certain graphs for each type of information-providing device. The way these devices work is simple; if the cop is provided with $\frac{1}{k}$ information, where k is an integer, then $\lfloor \frac{1}{k} \rfloor$ (note the floor function) of the vertices in G may have information devices placed on them. For example, if we have half information on some graph G, then up to half of the vertices in G may be occupied by information providing devices. We are focused on fixing the amount of information and determining the minimum number of cops that suffice to win because the complementary analysis (fixing the number of cops at one and determining the amount of information needed) has already been extensively studied. See, for example, [2, 3, 4, 6, 7].

This first chapter will continue to introduce the topic of Cops and Robbers, provide definitions for all applicable terms, and introduce and explain common notation. The second chapter will focus entirely on cameras. We will introduce the use of cameras and contrast our results with those for perfect information. Before generalizing our results to $\frac{1}{k}$ information, we will discuss results for half information. We start with half information because it gives us a sense of how cameras can be used and allows us to better understand the general results. We will also discuss some results for bipartite graphs and give some simple applications of these results. The third chapter will be much like the second chapter except the focus will be on alarms instead of cameras. We again contrast the results for alarms with those of cameras and full information. This will lead into us discussing results for half information before generalizing to $\frac{1}{k}$ information. This is useful again because it gives us a better understanding of how alarms work. The fourth chapter will be different from the rest of this thesis because we will instead fix the number of cops at one and determine the amount of information (alarms and cameras) needed. As previously stated, this version of the game has been well studied. However, during our research, we often compared the new strategies developed in this thesis to those that were previously known. As the underlying game is the same, the strategies are easily compared and ultimately, we include our results here because they improve on those results already known in [7]. The fifth and final chapter will examine the use of photo radar with and without direction. The work done in this chapter builds on and improves previous results from [6, 7]. We begin by fixing the information and determining the number of cops required. Then, given the number of cops required, we can go back and bound the information again (in most cases there is an excess but not enough to warrant fewer cops).

1.1 Definitions

We will now list some important graph theory definitions and notations that will be used throughout this thesis. See, for example, a standard graph theory text, such as [10]. A **graph** G is a set of **vertices**, $V(G)$, together with a set of **edges**, $E(G)$, which are two-element subsets of $V(G)$. An **edge** from vertex x to vertex y will be denoted xy, rather than $\{x, y\}$. Two edges are said to be **incident** if they share a common vertex and two vertices are said to be **adjacent** if they are connected by an edge, denoted by $x \sim y$. A **loop** is an edge which has the same endpoints and a graph is said to be **reflexive** if it has a loop at every vertex. The **degree** of a vertex is the number of edges incident with that vertex and it is denoted $\mathbf{deg}(v)$. We should note that loops are counted twice when finding the degree of a vertex. A **leaf** is a vertex with degree 1. Two vertices are said to be **independent** if there is no edge joining them. An **independent set** in a graph is a set of pairwise nonadjacent vertices. A **clique** in a graph is a set of pairwise adjacent vertices. A **subgraph** of a graph G is a graph H such that $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$ and the assignment of endpoints to the edges in H is the same as in G. $G - v$ is a subgraph of G obtained by removing the vertex v and all edges incident with v. An **induced subgraph** is a subgraph obtained by deleting a set of vertices. For the subgraph induced by T, we write $G[T]$ for $G - \bar{T}$, where $\bar{T} = V(G) - T$. An **isomorphism** from a simple graph G to a simple graph H is a bijection $f : V(G) \to V(H)$ such that $uv \in E(G)$ if and only if $f(u)f(v) \in E(H)$. If there is an isomorphism from G to H, we say "G is **isomorphic** to H." A graph G is **bipartite** if $V(G)$ is the union of two disjoint independent sets called **partite sets** of G. A **walk** is a list $v_0, e_1, v_1, ..., e_k, v_k$ of vertices and edges such that, for $1 \leq i \leq k$, the edge e_i has endpoints v_{i-1} and v_i. A $u, v - walk$ is a walk with u and v as the first and last vertex, respectively. These vertices are called **endpoints**. A **trail** is a walk with distinct edges (i.e. no edges are traversed more than once). A **path** is a walk with distinct vertices. If a walk is **closed** the first and last vertices are equal. A **cycle** is a closed trail with distinct vertices and an n**-cycle** is a cycle with n vertices. A graph, G, is **connected** if there exists a $u, v - path$ for every $u, v \in V(G)$ (otherwise, G is disconnected). A graph with no cycle is **acyclic**. We will now introduce some important classes of graphs that will be referenced throughout this thesis.

A **wheel graph,** W_n**,** is a graph with $n \geq 4$ vertices and is formed by connecting a single vertex to all vertices of an $(n-1)$-cycle. The middle vertex is adjacent to all the outer vertices and an eight vertex wheel graph is shown in Figure 1.1.

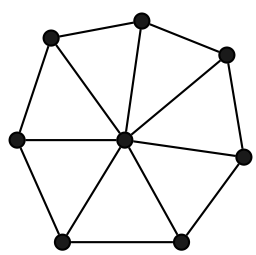

Figure 1.1: The wheel graph on 8 vertices, W_8

A **complete graph,** K_n, is an n-vertex simple graph where every vertex is adjacent to every other vertex. An example is shown in Figure 1.2.

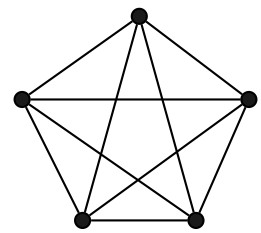

Figure 1.2: K_5

A **tree** is a connected acyclic graph. A **rooted tree** is a tree with one vertex, r, chosen as the **root**. Visually, the **root** of a graph is the topmost vertex. For example, in Figure 1.3, the topmost vertex is the root. Any vertex can be chosen as the root because different drawings or representations of a graph are isomorphic. For each vertex, v, let $P(v)$ be the unique $v, r - path$ in a tree, T. The **parent** of v is its neighbour on $P(v)$ and its **children** are its other neighbours. A **subtree** of a tree T is a tree consisting of a vertex in T (the root of the subtree) and all of its descendants in T. The **depth** of a tree is the largest distance between the root and any of its leaves. A **full complete k-ary tree** is a tree in which all vertices except the leaves have k children and the distance between any leaf and the root is the same. For example, Figure 1.3 shows a full complete ternary (3-ary) tree of depth 2.

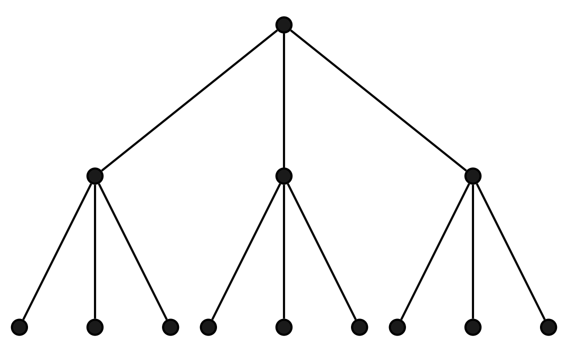

Figure 1.3: Ternary tree of depth 2

We now consider graph products and more specifically, the strong product, lexicographic product and Cartesian product. It should be noted here that we can take the product of any two graphs; however, in this thesis, the graph product we use will always be the product of two paths. This restriction allows for easier notation and when we refer to a grid it will be as "an $m \times n$ (type of product) grid."

The **strong product**, $G \boxtimes H$, of two graphs G and H is defined such that the vertex set of $G \boxtimes H$ is the Cartesian product of $V(G) \times V(H)$ and any two vertices (u, u') and (v, v') are adjacent in $G \boxtimes H$ if and only if u' is adjacent to v' (or $u' \sim v'$) and u is adjacent to v (or $u \sim v$). An example of a 3×3 strong grid is given in Figure 1.4.

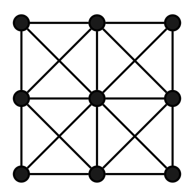

Figure 1.4: 3×3 strong grid

The **Cartesian product**, $G \square H$, of two graphs G and H is defined such that the vertex set of $G \square H$ is the Cartesian product of $V(G) \times V(H)$ and any two vertices (u, u') and (v, v') are adjacent in $G \square H$ if and only if either $u = v$ and u' is adjacent to v' in H, or $u' = v'$ and u is adjacent to v in G. An example of a 3×3 Cartesian grid is shown in Figure 1.5.

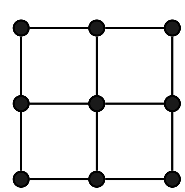

Figure 1.5: 3×3 Cartesian grid

The **lexicographic product**, $G \cdot H$, of two graphs G and H is defined such that the vertex set of $G \cdot H$ is the Cartesian product of $V(G) \times V(H)$ and any two vertices (u, u') and (v, v') are adjacent in $G \cdot H$ if and only if either u is adjacent to v in G, or $u = v$ and u' is adjacent to v' in H. An example of a 3×3 lexicographic grid is shown in Figure 1.6.

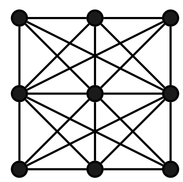

Figure 1.6: 3×3 lexicographic grid

Next we will introduce some definitions, terms and results specific to the Cops and Robber game. The rules are discussed in Section 1.2. A graph is said to be **copwin** if one cop suffices to catch the robber. A graph is said to be **robberwin** if one cop cannot guarantee the capture of the robber after a finite number of moves. A graph is n**-copwin** if n cops can apprehend the robber. A k**-copwin band** is an $m \times n$ grid in which k cops can apprehend the robber with no information. For example, a path is a 1-copwin band and a $2 \times n$ Cartesian grid is a 2-copwin band. More simply, a copwin band is a grid in which the cop(s) begin at one end and move towards the other until the robber is apprehended. The robber has no method of escape unless he moves off the copwin band and onto an information providing device. A **free** vertex is a vertex without a detection device (we will use this interchangeably with information providing device) placed on it. A **free** edge is an edge without a detection device on it. A path P is said to be a **freepath** if every edge of P is free.

Lastly, we define what **searching** means in this thesis. In the graph theory literature, searching generally refers to the case of continuous movement of the cop and robber. See, for example, [1]. In this variant of Cops and Robber, the cop and the "fast" robber can move continuously throughout the graph. However, we will use the word "search" in a different way and the main difference in the definitions is that in our case, the moves made by the cop and robber are discrete (i.e. from one vertex to another). We define **searching** as the case when the cop(s) exhaustively check each vertex in a section of a graph (usually a copwin band) or an entire graph for the robber. Furthermore, searching will imply that the robber has no information. For example, we would say that the cop searches a path for the robber or that the two cops search a 2-copwin band until the robber is apprehended.

1.2 Rules of the Game

Before we begin discussing the copwin characterization, we should outline the basic rules of the original game. First, suppose G is a finite, reflexive, connected graph. The original game is played by one cop and one robber. The cop begins by choosing a starting vertex of G, after which the robber chooses a starting vertex. The cop and robber alternate moves (starting with the cop), either moving along an edge to a new vertex or moving along a loop (called "passing") and staying at their current vertex. Each player can move once per turn and the object of the game is for the cop to apprehend the robber after a finite number of moves. As discussed above, this occurs when the cop occupies the same vertex as the robber. In this version of the game, both sides play with perfect information, meaning that they know each other's location at all times.

1.3 Copwin Characterization

As previously mentioned, the original game was played with only one cop and one robber. This allowed all graphs to be characterized as either copwin or robber-win. Copwin graphs were characterized in [8] and [9]. Before we proceed with the copwin characterization, we need some specific definitions. Let G be a graph and let $v \in V(G)$. The **neighbourhood** of v, denoted by $N(v)$, is the set of vertices adjacent to v. The **closed neighbourhood** of v, denoted by $N[v]$, is given by $N(v) \cup \{v\}$. A vertex d of a graph G said to **dominate** another vertex v if d is adjacent to each of the vertices in the closed neighbourhood of v.

Suppose we have a copwin graph. We know that after a finite number of moves the robber must be apprehended, so let's consider the robber's last move. He can either pass and stay on his current vertex, move to the vertex the cop is occupying, or move to a vertex adjacent to his current vertex. In all of these cases, the vertex that the robber is on or moves to must be adjacent to the vertex the cop is occupying. This must be true because we assumed that the robber is about to lose. In other words, the vertex that the cop is occupying, call it d, must dominate the vertex that the robber is occupying, call it v. We then say that v is a **corner** since, once the robber moves onto v, he has no method of escape. Clearly, a graph without a corner is not copwin. However, a graph with a corner is not immediately copwin because we need to be able to force the robber into occupying that corner vertex. We can think about this as removing the corner vertex, v, from G and examining if $G' = G - v$ is copwin. If G' is copwin, then it must also have a corner. We again remove the corner and examine if the remaining graph has a corner. We continue to dismantle the graph until only a single vertex remains. The ordering in which we remove the corners is known as a **copwin ordering**. The vertices and edges that are part of the copwin ordering are known as the **copwin spanning tree**. If it is possible to dismantle G in this way, then it is characterized as copwin. This characterization actually holds in both directions (see [8]) and if a graph, G, is characterized as copwin, then it is possible to dismantle G as described above. Let's look at the example in Figure 1.7, which shows a copwin graph, G. Clearly, a is a corner and it is dominated by b, meaning that we can remove a. Now we are left with $G' = G - a$ and b is now a corner dominated by c. We remove b and the remaining graph has another corner c, dominated by f. In fact, e and d are also corners, dominated by f. Removing c, d, and e, we are left with a single vertex f. Thus our graph is copwin. The copwin ordering we used was (a, b, c, d, e, f) and we should note that this ordering is not unique. We could have similarly used the copwin ordering (a, b, d, e, c, f) or (f, e, d, c, b, a).

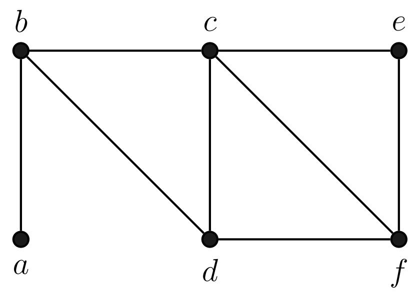

Figure 1.7: Copwin graph with copwin ordering (a, b, c, d, e, f)

1.4 Copwin Strategy

Now that we have shown how to characterize a graph as copwin, we need to discuss the strategy used by the cop to actually apprehend the robber. The copwin strategy and the proof that it always works can be found in [3].

Suppose that G is a copwin graph and that $(x_1, x_2, ..., x_n)$ is a copwin ordering of the vertices of G. Define the subgraphs $G_i = G_{i-1} - x_i$ where $G_1 = G$. The robber is always thought of as playing the game on the original graph G. However, the cop is thought of as playing the game on G_i, $i = 1, 2, ..., n-1$. We can think of the G_i's in terms of how we removed the corner vertices. Instead of removing a corner vertex, we are actually mapping it onto the vertex that dominated it. Such a mapping is called a **one point retraction** since all other vertices are mapped to themselves. We let $f_i : G_i \to G_{i+1}$ be the mapping which maps x_i onto a vertex that dominates it. This allows the cop to pursue the robber's image on each G_i, eventually apprehending him on G. The cop begins by starting on the vertex x_n and thus initially moves on G_n. When the cop occupies the robber's image on G_i, he then moves onto the image of the robber in G_{i-1}. This is always possible because of the properties of the retraction mapping. More simply, if the cop is occupying the robber's image on G_i then any vertex to which the robber can move is adjacent to the cop's vertex in G_{i-1}. After at most n moves, the robber is apprehended.

Let's look at the example presented in Figure 1.7. The copwin ordering is (a, b, c, d, e, f) and so the cop begins by choosing vertex f. Let's assume the robber chooses vertex b. Through the retraction mapping, the cop is currently occupying the robber's image on G_6 and so he moves onto the image of the robber on G_5, which is still vertex f. We should note here that G_5 is the subgraph of G induced by the vertices e and f. Now, let's assume the robber passes. Since the cop is still occupying the robber's image, he moves onto G_4 and vertex d. If the robber moves to vertex a, the cop is still occupying his image and moves to vertex c in G_3. The cop continues moving through the copwin ordering in this manner until he apprehends the robber.

1.5 Original Game Results

Before we proceed, we should look at some results and examples from the original game. The simplest example of a copwin graph is a path. The cop will start on one of the two end vertices and move down the path towards the other end. Clearly, the robber cannot escape and will eventually be apprehended. Thus, all paths are copwin. Other easy examples of copwin graphs are complete graphs and wheel graphs. For complete graphs, the cop may start on any vertex and apprehend the robber in exactly one move. This is because each vertex in a complete graph is adjacent to every other vertex. For a wheel graph, the cop will start on the middle vertex (which is also adjacent to every other vertex) and again, apprehend the robber after one move. Cycles are different because any cycle with four or more vertices is robber win. For example, let's consider a 4-cycle (i.e. a square). The cop will chose a starting vertex and the robber will chose the opposite corner as his starting vertex. The cop can never catch the robber because the robber can always stay one vertex ahead of the cop. Trees are another example of a class of copwin graphs. The cop starts on the root and moves down the tree towards the robber. In doing so, the cop continuously shrinks the size of the tree that they are playing on. By definition there is only one path between any two vertices on a tree and this means the robber must continuously move down the tree (away from the cop), towards the leaves. Eventually, the robber will land on a leaf and the cop will apprehend him there, if not before.

Next, we need to look at the results for perfect information on strong, Cartesian and lexicographic grids. Let's begin with strong grids. In order to claim that strong grids are copwin, we must show that they dismantle as described in Section 1.3. Suppose we have an $m \times n$ strong grid. The vertices in row m (or column n) are dominated by the vertices in row $m-1$ (or column $n-1$). For example, in Figure 1.8, vertex d dominates vertices a, b, and c. In this way, we can continue to remove vertices until we are left with a single vertex. Thus, strong grids are copwin with one cop.

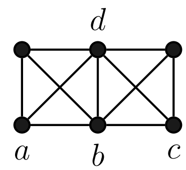

Figure 1.8: 2×3 strong grid

Next, let's consider lexicographic grids. Suppose we have an $m \times n$ lexicographic grid. A vertex in a given row is dominated by the vertex that precedes it in the same row. For example, in Figure 1.9, vertex a is dominated by vertex b and vertex b is dominated by vertex c. In this way, each row can be mapped to a single vertex. This leaves us with a single column (or path) of vertices which can clearly be dismantled and thus lexicographic grids are copwin.

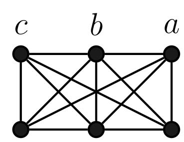

Figure 1.9: 2×3 lexicographic grid

Finally, let's consider Cartesian grids. Suppose we have an $m \times n$ Cartesian grid. Cartesian grids are not copwin with one cop and cannot be dismantled like in Section 1.3. They are not copwin because, as we discussed earlier, a 4-cycle is not copwin. However, a Cartesian grid is 2-copwin. The search strategy is as follows:

Both cops begin on adjacent vertices, one in row m and one cop in row $m-1$. The cops move vertically towards the row that the robber is occupying until they are in the same row or the robber begins to move vertically. Once one of the cops is in the same row as the robber they move horizontally and force the robber to move vertically. The cops will move vertically with the robber until he reaches the end of the graph. At this point the cops continue to move vertically (even while the robber is moving back towards them) until one of them is in the same row as the robber. The cops will then move horizontally until the robber is no longer in one of their rows. We repeat this process, continuously decreasing the distance between the cops and the robber, thus eventually apprehending the robber.

This search strategy is used though out the theses and is formalized in the next section.

1.6 Search and Placement Strategy

We now present an important theorem that will be used numerous times throughout this thesis. This theorem will be used to prove that we can always apprehend the robber using a similar search and placement strategy with all different types of information. Before we begin, some clarification about the use of "we" is needed. During many of the proofs and throughout this thesis, "we" will be used synonymously with the cop(s).

Strategy 1: We are given a graph G, which contains only the vertices (and no more) of an $m \times n$ Cartesian grid. The exact description of G will be given in each individual section. More simply, G has the vertices and at least the edges of an $m \times n$ Cartesian grid. Depending on the section, G may be a lexicographic or strong grid; these are simply Cartesian grids with added cross edges. In each case, we are given a preset amount of information in the form of one of cameras, alarms and photo radar with or without direction. The exact amount of information and cops is given in each case. However, our information is always arranged so that we have horizontal copwin bands and so that we know if the robber attempts to move vertically in our graph.

Theorem 1. *Strategy 1 can be used to show that G is copwin.*

Proof. The proof is broken down into two cases as follows:

Case 1: The robber never triggers a detection device.

The cop(s) search(es) the copwin bands until the robber is apprehended.

Case 2: The robber triggers a detection device.

First, we should note that the robber can no longer get behind the cop(s) because of how we've arranged our detection devices.

Now, the cop immediately moves vertically (or diagonally if possible) towards the robber. If the robber attempts to move horizontally, he is either on a detection device or in a copwin band. In either case, we know what row he is in and the cop will continue to move vertically until he is in the same row as the robber or until the robber moves vertically. Once the cop is in the same row he can search horizontally and force the robber to move vertically.

The cop will continue to move vertically until the robber reaches the end of the graph. At this point the cop continues to move vertically (even while the robber is moving back towards him) until he is in the same row as the robber. He will then move horizontally until the robber is no longer in our row.

We repeat this process, continuously decreasing the distance between the cop and the robber, thus eventually apprehending the robber. □

1.7 Partial Information Results

Before we proceed to Chapter 2, we need to outline the main known results for Cops and Robber with partial information that will be used in this thesis, which can be found in [2, 3, 4, 6, 7]. These results fix the number of cops at one and give bounds on the amount of information needed. The placement strategy is best explained using photo radar on trees and extended to other information devices and copwin graphs from there. Let T_a be a tree T rooted at vertex a. Define an **a-branch** of T_a to be a path of T with a as one end vertex. The idea is that we place the photo radar units in such a way that the free edges form freepaths and each maximal freepath is on an a-branch. On trees this is simple; at each vertex we leave the left most edge free and place photo radar units on the others. Edges incident with leaves do not receive photo radar units. Note that this applies to both photo radar units with and without direction. Let's consider the example in Figure 1.10. The thick edges represent the placement of photo radar units.

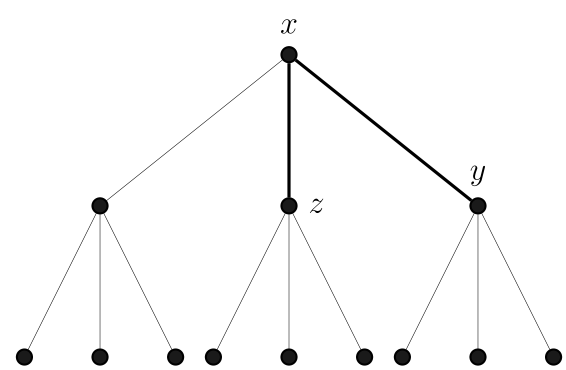

Figure 1.10: Ternary tree of depth 2

Extending this to other information providing devices is straightforward. We know that a camera placed at a vertex, x, can provide the same information as any photo radar units (on edges) emanating from that vertex. Let $xy \in E(T)$ with x as the parent of y (or equivalently, x is after y in the copwin ordering) be an edge of T with photo radar. The photo radar unit can be replaced by a video camera on vertex x (this ensures we never place a camera on a leaf). If we look at Figure 1.10, the camera will be placed on vertex x to replace both photo radar units. Lastly, we consider alarms. At each vertex x we leave the leftmost child free and place an alarm on the others. Again, we do not place alarms on leaves. In Figure 1.10, alarms are placed on vertices x, y, and z.

Finally, we must extend these results to copwin graphs and this is straightforward because of the copwin spanning tree. For copwin graphs, we place information devices as we just discussed on the copwin spanning tree. The edges not contained in the copwin spanning tree all receive information providing devices (provided we are using edge detection devices).

Chapter 2

Cameras

2.1 Partial Information

We begin by examining the use of cameras on two classes of graphs that provide insight in showing us the vast difference between partial information and full information. Both classes of graphs, wheel graphs and complete graphs, are copwin in one move with full information. However, once the amount of information is restricted, things are much different. We begin with wheel graphs.

Theorem 2.1.1. *Given a wheel graph with n vertices, the minimum number of cameras needed for one cop to guarantee a win is two.*

Proof. One camera is placed in the middle and one on any of the outer vertices. This creates a path for the cop to search. The cop begins at one end, next to the camera and moves around the outside of the wheel. If at any time the robber moves into the middle, he is apprehended. As soon as he moves onto the camera not in the middle, the cop moves to the middle and the robber is apprehended on the next move. If the robber never moves onto a camera, he will be caught by the cop searching the outer vertices, a path. □

As we can see, the wheel graph requires very little information in order for the robber to be apprehended. This might be expected since it is very easy for one cop to win with full information. However, in contrast to wheel graphs, we will now examine complete graphs. Complete graphs are also very easily won with one cop and full information. Let K_n be a complete graph on n vertices, let A be the proportion of vertices in K_n without cameras, let a be the proportion of vertices in K_n with cameras, let c be the number of cameras and let k be the number of cops. We can see that $A = 1 - a$ and $A \cdot n = n - c$. We want to know, given any complete graph, K_n, and c cameras, how many cops are needed to guarantee a win.

Theorem 2.1.2. *Given a complete graph with n vertices and c cameras, the minimum number of cops needed to guarantee a win is* $k = \lceil \frac{n-c}{2} \rceil$.

Proof. Our first case is if the robber begins on a vertex with a camera. He is immediately apprehended on the first move. Suppose that the robber begins on one of the other $n - c$ vertices and assume that the cops do not start on the vertices with cameras. Thus, of the remaining $n - c$ vertices, our relationship tells us that at least half are occupied by cops. On the first move, all the cops will move simultaneously to a distinct free (without a camera) vertex and thus apprehend the robber. If there are more cops than free vertices, some cops may move to the same free vertex.

Now we must show that we cannot do better (i.e. use fewer cops). Let's assume that we have $k = \lceil \frac{n-c}{2} \rceil - 1$ cops and show this leads to a robber-win scenario. We now know that we have one or two less than half of the free vertices occupied by cops. This means that when all the cops simultaneously move to a distinct free vertex, one or two vertices will remain and if the robber occupies either of these vertices, he will not be apprehended. On the next move, the robber moves (or stays) to another free vertex and when the cops again move to distinct free vertices there will be one remaining. Although the probability of the robber continuously choosing the correct vertex each time will be small, it means we cannot *guarantee* his capture. □

Of course, we could have more cops than needed, in which case more than half of the free vertices will be occupied by cops. This case will allow some cops to remain on their respective vertices. This relationship also tells us that, on complete graphs, two cameras are worth one cop to a minimum of one cop remaining. This is because in our scheme, 1 cop is responsible for two vertices, the vertex he begins on and the vertex to which he moves. A camera is only capable of occupying one vertex and so we would need two cameras to occupy both of the two vertices that 1 cop can cover. However, since cameras cannot capture the robber, we must have at least 1 cop remaining in order to apprehend the robber.

Even though complete graphs are one of the best cases with full information, it is clear that they are one of the worst cases with partial information. In fact, we have shown that at least half of the vertices must have either a cop or a camera. The vast difference between how wheel graphs and complete graphs transition to partial information illustrates the difficulties in characterizing all copwin graphs with partial information. As we can see, with partial information each class of graph may differ greatly with the amount of information needed and it may not correspond at all to the results with full information. That being said, we proceed to a large class of similar graphs.

2.2 Grids

All of the grids referred to in this thesis are created using two paths of length m and n. The three main types of grids on which we will focus are: strong, Cartesian and lexicographic grids. Each type of grid will be examined fully for each type of detection device. We can draw similar results from this large class of graphs because each of these grids is very similar in its underlying appearance. This also allowed us to come up with Theorem 1 which will be used extensively in this section.

Strong grids are a good place to begin because they are copwin with one cop and full information. One of the first questions that should be asked is, "Is full information necessary for one cop to win or can it be done with less?" It turns out half information is sufficient and is discussed below.

Theorem 2.2.1. *Given an $m \times n$ strong grid with half information provided by cameras, one cop can guarantee a win.*

Proof. We begin the proof with the strategy for placing cameras. In the future, this will be denoted with **PS** (placement strategy). Orient the graph so $m \leq n$. Place cameras across rows 2 and $m-1$. Alternate rows in between them and never leave two consecutive rows empty. Cameras should not be placed in row 1 or row m because we are placing cameras in rows 2 and $m-1$. It is more efficient to place the cameras in rows 2 and $m-1$ because the camera will tell us if the robber has gone into the outermost row and, since it is a copwin path, it can be searched by one cop. Any extra cameras should be placed in the centre of an empty row, starting with rows closest to $\frac{m}{2}$. Note that any extra cameras are not needed and can be placed anywhere. No 3-cycles should remain at this point because alternating rows have cameras.

We know that the relationship $\frac{m}{2} \geq \lfloor \frac{m}{2} \rfloor$ will hold for any m. This confirms that with half information we will be able to, at the least, have cameras in alternating rows. Also, notice that if the number of rows is odd, we will have cameras remaining to place in the middle of empty rows. We now proceed with the proof that the cop can apprehend the robber or the search strategy. In the future, this will be denoted **SS** (search strategy). The proof of this follows directly from Theorem 1.

Start at position $m = 1, n = 1$ (i.e. the upper left corner of the graph). Search across row $m = 1$.

Case 1: If the robber never lands on a camera, continue searching empty rows (i.e. $1, 3, 5, ..., m$) until the robber is apprehended.

Case 2: If the robber attempts to enter the row we are searching behind us, we will know his point of entry since alternating rows have cameras. We immediately move towards him until he is apprehended or until he moves up or down a row. If he moves up or down, he must be moving onto a camera and we move diagonally into the robber's row. If he moves into a row without a camera we can also move into that row diagonally and move down the row towards him. We can now guarantee that he will never get behind us. This allows us to continually follow him and the distance between us will be non-increasing and eventually decreasing since the graph is finite. The distance between the cop and the robber will be decreasing because of the finite nature of the grid. Once the robber reaches the end of the grid, he must turn around. Since the cop is following behind him, the robber must pass the cop and this allows the cop to decrease the distance between them at each turn.

Case 3: The robber triggers a camera on the graph and moves into an empty row. We choose a direction and move diagonally so as to intercept the robber. If we choose the right direction we will either apprehend him or end up in front of him and then we can proceed as in Case 2. If we choose the wrong direction, he will be in the same row as us and we can again proceed as in Case 2. □

Before we proceed, let's consider the example in Figure 2.1. All vertices in rows 2 and 3 receive cameras. The cop begins on the leftmost vertex in row 1. He will search for the robber as described above.

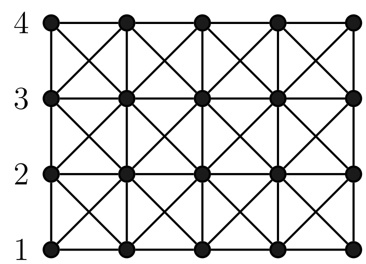

Figure 2.1: 5×4 strong grid

It turns out that for certain strong grids, half information is just enough and yet in others, we have extra cameras available. That being said, half information appears to be a good bound for one cop on strong grids. We now turn to Cartesian grids which are not copwin with one cop and full information, but are copwin with two cops and full information. Taking the same approach, we examine what happens with only half information.

Theorem 2.2.2. *Given an $m \times n$ Cartesian grid with half information provided by cameras, two cops can guarantee a win.*

Proof. **PS:** Orient the graph so $m \leq n$. Place the cameras across every third row starting with row 1. The number of extra cameras is given by $\lfloor n(\frac{m}{2} - \lceil \frac{m}{3} \rceil) \rfloor$. We place the extra cameras in the middle of the empty rows starting with the rows closest to $\frac{m}{2}$. We notice that the number of extra cameras gets larger and larger as the graph increases.

SS: The proof follows directly from Theorem 1.

The two cops will search side by side covering two rows at a time staring at one end of rows 1 and 2 (i.e. m=1 and m=2 with n=1).

Case 1: If the robber never lands on a camera, we search the empty rows until he is apprehended.

Case 2: The robber attempts to get behind us. As soon as he triggers a camera behind us, we make one move in that direction.

a) If he moves into our row, we continue to move towards him until he enters a different empty row, then we move vertically. This means we are one row behind him moving vertically.

b) If he doesn't move into our row but a row above or below, we move vertically right away.

In the worst case, he only moves vertically and we can chase him to the end of the grid and move closer at each end, eventually catching him. □

Another 2-copwin strategy for Cartesian grids with half information is as follows. Place cameras across the first row on alternating vertices starting with the vertex in column two. Place cameras across the second row starting with the vertex in column one. Continue alternating in this manner. This will create a dominating set; every vertex has a camera or if not, every adjacent vertex has a camera. We implement the same search method as above except we now know whenever the robber makes a move. This creates the illusion of perfect information and since an $m \times n$ Cartesian grid is 2-copwin with perfect information, it is 2-copwin with half information.

The lexicographic is similar to the strong grid in that it is copwin with one cop and full information and the results with half information are essentially the same.

Theorem 2.2.3. *Given an $m \times n$ lexicographic grid with half information provided by cameras, one cop can guarantee a win.*

Proof. **PS:** Orient the graph so that the long edges are horizontal. Place cameras across rows 2 and $m-1$. Alternate rows in between them and never leave two consecutive rows empty. Also, cameras should not be placed in row 1 or row m. Any extra cameras should be placed in the centre of empty rows, starting with rows closest to $\frac{m}{2}$. No 3-cycles should remain at this point. This is very similar to that of strong grids.

SS: The proof of this follows directly from Theorem 1 and is exactly the same as the strong grid case. □

As we have seen, all three types of grids are copwin with half information and require the same number of cops as with perfect information. An interesting extension is the Cartesian product of three paths. This three dimensional graph is also winnable with two cops and half information as shown below.

Theorem 2.2.4. *Given the Cartesian product of three paths with half information provided by cameras, two cops can guarantee a win.*

Proof. First, let j, m, n be the lengths of three paths.

PS: Orient the graph so that $j \geq m$ and $n \geq m$. The nj face corresponding to $m = 1$

will have two orientations that place the cameras on alternating vertices. Choose the method that uses fewer cameras. For the nj face corresponding to $m = 2$, use the other method and continue to alternate methods in this manner. This will guarantee that every vertex is either occupied by a camera or if not, then all the adjacent vertices have cameras. This arrangement with tell us every time the robber moves.

SS: Start on two empty vertices in a four cycle (i.e. two opposite corners of a four cycle) on the nj face corresponding to $m = 1$. Move to the nj face corresponding to $m = 2$ (think of this as moving into the page). Now, move to the other two vertices on the four cycle (i.e. the other two corners of the four cycle). Continue searching in this way until the robber triggers a camera. If he never triggers a camera, eventually we will apprehend him. As soon as he triggers a camera, we will know his every move and thus it becomes a game of perfect information. □

As mentioned above, we can create the illusion of perfect information by placing cameras in a specific manner; this only occurs in bipartite graphs. The following theorem follows directly from the idea developed above.

Theorem 2.2.5. *Given an n-dimensional Cartesian grid with half information provided by cameras, n cops can guarantee a win.*

Proof. We label the vertices as points with n coordinates with the bottom left vertex receiving all zeroes (i.e. the origin). We then partition our vertices into two sets. Let A be the set of vertices with an even coordinate sum and let B be the set of vertices with an odd sum. We know that two vertices are adjacent if and only if they differ by one coordinate and so we have no edges between vertices in sets A or B. Thus we have a bipartite graph. We choose to place cameras on all vertices in either set A or

B. This means that when the robber moves he is either moving from a camera to an empty vertex or vice versa. This essentially creates a graph with perfect information and is therefore n-copwin. $\square$

The previous theorem uses a camera placement strategy that directly depends on the graph being bipartite and before we generalize this result for all bipartite graphs and comment on its applications, we must first conclude our investigation into grids. Thus far, we have only discussed grids with half information. The obvious next step is to generalize the results for $m \times n$ Cartesian, strong, and lexicogaphical grids. We want to determine the number of cops required for all given levels of information (i.e. with $\frac{1}{k}$ information).

Theorem 2.2.6. *Given an $m \times n$ strong or lexicographic grid with $\frac{1}{k}$ information provided by cameras, $k-1$ cops will suffice, with $k \geq 2$, to apprehend the robber.*

Theorem 2.2.7. *Given an $m \times n$ Cartesian grid with $\frac{1}{k}$ information provided by cameras, $k-1$ cops will suffice, with $k \geq 3$, to apprehend the robber.*

Proof. First we should note that the camera placement strategies for strong, Cartesian and lexicographic grids are the same as in Theorem 2.2.1, Theorem 2.2.2 and Theorem 2.2.3, respectively. From those strategies, we know that $\left\lfloor \frac{\min(m,n)}{k} \right\rfloor$ is a bound on the complete rows of cameras. This implies that we will need a total of $\left\lfloor \frac{\min(m,n)}{k} \right\rfloor \max(m,n)$ cameras for this strategy. We need to ensure that the number of cameras available is always larger than or equal to the number of cameras we need.

We need to show the following inequality holds for any m, n and k.

$$\left\lfloor \frac{mn}{k} \right\rfloor \geq \left\lfloor \frac{\min(m,n)}{k} \right\rfloor \max(m,n)$$

Let $m = \min(m,n)$, $n = \max(m,n)$ and $a = \frac{m}{k}$. We now have to show

$$\lfloor an \rfloor \geq \lfloor a \rfloor n.$$

Using Hermite's Identity

$$\lfloor an \rfloor = \sum_{n=0}^{n-1} \left\lfloor a + \frac{k}{n} \right\rfloor$$

and

$$\lfloor a \rfloor n = \sum_{n=0}^{n-1} \lfloor a \rfloor .$$

And so we conclude that

$$\sum_{n=0}^{n-1} \left\lfloor a + \frac{k}{n} \right\rfloor \geq \sum_{n=0}^{n-1} \lfloor a \rfloor .$$

$\square$

Now we need to show the number of cops required by this strategy. We should note here that given an $m \times n$ grid with $m \leq n$ and no camera's, m cops can search the grid. This is the simplest case and the idea is that we have one cop per row and they move in tandem across the grid until the robber is apprehended. Now extending this idea, we first take the number of rows and divide it by the number of rows with cameras and call the result J. This will tell us that there is a camera every J rows. We subtract 1 from J and we will get the number of consecutive empty rows (i.e. the number of cops required). The number of rows is given by: $\min(n,m)$. The number of rows with cameras is given by: $\left\lfloor \frac{\min(n,m)}{k} \right\rfloor$. The number of cops required is split into two cases.

Case 1: $\left\lfloor \frac{\min(n,m)}{k} \right\rfloor = 1$. The number of cops is given by:

$$\left\lfloor \frac{\min(n,m)}{2} \right\rfloor$$

Case 2: $\left\lfloor \frac{\min(n,m)}{k} \right\rfloor \neq 1$. The number of cops is given by:

$$\left\lfloor \frac{\min(n,m)}{\left\lfloor \frac{\min(n,m)}{k} \right\rfloor} \right\rfloor - 1$$

$$= \left\lfloor \frac{k \min(n,m)}{\min(n,m)} \right\rfloor - 1$$

$$= k - 1.$$

Now that we have generalized our results for strong, Cartesian and lexicogaphical grids, we can confirm that our results are the same as we found using half information. The number of cops is kept low because cameras are the most useful of the detection devices in the sense that one camera provides more information than one of any other device.

2.3 Bipartite Graphs

We now examine bipartite graphs with half information provided by cameras.

Theorem 2.3.1. *Every bipartite graph G, that is n-copwin with perfect information is also n-copwin with half information using cameras.*

Proof. Suppose we have our two partite sets A and B where $A \leq B$. We know that $|A| \leq \frac{1}{2}|G|$. Place cameras on the vertices in A. Now every move the robber makes is from a vertex with a camera to an empty vertex or visa versa. Since we have direction, we have created perfect information once the robber first moves onto a camera. If the robber never moves, we search the empty vertices until he is apprehended.

□

A few simple applications of this theorem to specific classes of graphs are discussed below. We should note that these results follow easily from the results presented in Section 1.7.

Theorem 2.3.2. *Given an n-vertex tree with half information provided by cameras, one cop can guarantee a win.*

Proof. All trees are bipartite since they have no odd cycles and a tree with perfect information is copwin with one cop. Theorem 2.4.0.1 applies. □

Theorem 2.3.3. *A star graph with n vertices is always copwin with half information provided by cameras and one cop. In fact, a star graph is copwin with one cop and one camera.*

Proof. First, we know that since the star graph is bipartite, $K_{1,n}$, we know that it will be copwin with one cop and half information by Theorem 2.4.0.1. We note that we can refine this result and use only one cop and one camera. Place the camera in the middle. Search the outer vertices. If the robber attempts to get behind us, he moves onto the middle where he is apprehended. □

As we can see, the results for bipartite graphs depend entirely on the results already known for the underlying graph with perfect information and it is essentially an easy extension of previous results. We have finished generalizing our results for cameras given $\frac{1}{k}$ information and we now proceed to alarms.

Chapter 3

Alarms

3.1 Partial Information

Alarms and cameras are very similar in that they are both detection devices placed on vertices, however, the loss of direction makes alarms much less informative then cameras. Two alarms per edge are needed to equate the information provided by one camera and, because of this, the cop-number for many graphs is likely to increase, especially when holding the amount of information constant.

We begin our investigation into alarms much like we did with cameras, comparing two similar classes of easily copwin graphs.

Theorem 3.1.1. *Given a wheel graph with n vertices, the minimum number of alarms needed for one cop to guarantee a win is three.*

Proof. One alarm is placed on the middle vertex and the other two are placed on any two adjacent outer vertices. This creates a freepath for the cop to search. The cop begins at one end, next to the alarm and searches the path. If at any time the robber moves into the middle, he is apprehended. As soon as he moves onto either alarm not in the middle, the cop moves to the middle. The robber now has 3 options. If he remains on his current vertex, he is apprehended. If he moves to the adjacent vertex with an alarm he is apprehended. Finally, if he moves to the vertex he was previously on, he is apprehended. This is because we know he left the vertex he was on and if he did not trigger another alarm then he must have moved back to the previous vertex and thus we apprehend him. If the robber never triggers an alarm, he will be caught by the cop searching the 1-copwin path. □

As we can see, wheel graphs require only one more alarm than cameras. As it turns out, the results for complete graphs with alarms are exactly the same as the results with cameras. *Given a complete graph with n vertices and k alarms, the minimum number of cops needed to guarantee a win is* $c = \lceil \frac{n-k}{2} \rceil$. Again, we can see that complete graphs require a large amount of information and many cops in order for the robber to be apprehended. This is in contrast to wheel graphs, which again require very little information and only one cop. This further illustrates how differing classes of graphs may have very different results with less than full information.

3.2 Grids

As we did in the previous section, we will begin with grids and half information. This allows us to get a feel and understanding for the way alarms work before generalizing our results to $\frac{1}{k}$ information. The searching and alarm placement strategy is very similar to that of cameras. We use the fact that placing alarms in consecutive rows will essentially give us directional information and this allows us to use Theorem 1 in proving that the robber will eventually be apprehended. We begin with strong grids:

Theorem 3.2.1. *Given an $m \times n$ strong grid with half information provided by alarms, two cops can guarantee a win.*

Proof. **PS:** Orient the graph so $m \leq n$ and place alarms across rows 3 and 4. Continue placing alarms in consecutive rows skipping two rows between (i.e. place alarms on every 3rd and 4th row). Place any extra alarms across the rows closest to $\frac{m}{2}$. We place alarms in consecutive rows because it allows us to know the vertical direction that the robber moves and two consecutive rows of alarms act similar to one row of cameras. We will know if he enters an empty row because consecutive rows have alarms and so moving off an alarm out of vision means he entered an empty row (and of course we will know which one).

SS: This follows directly from Theorem 1.

The cops will search side by side searching two rows at a time. Start at position $m = 1, n = 1$ and $m = 2, n = 1$ (i.e. the upper left corner of the graph). Search across rows $m = 1$ and $m = 2$.

Case 1: If the robber never lands on an alarm, continue searching empty rows until the robber is apprehended.

Case 2: If the robber attempts to enter the row we are searching behind us, he will trigger an alarm two rows below us. We immediately move diagonally toward him with both cops. If he moves up or down, we move diagonally into the row he is in (only one of the cops is in his row). If he moves into a row without an alarm, we can also move into that row diagonally and move down the row towards him. We can now ensure that he will never get behind us. This allows us to continually follow him and the distance between us will be non increasing and eventually decreasing since the graph is finite.

Case 3: If the robber triggers an alarm on the graph and moves into an empty row. We choose a direction and move diagonally as to intercept the robber. If we choose the right direction we will either apprehend him or end up in front of him and then we can proceed as in Case 2. If we choose the wrong direction, he will be in the same row as us and we can again proceed as in Case 2. □

Notice that with alarms, an extra cop was added in order for the robber to be apprehended with half information. This is because by using the alarms in two consecutive rows, we leave two consecutive rows unalarmed and two cops are needed to search these copwin bands. The results for the lexicographic grid follow directly from the strong grid results.

Theorem 3.2.2. *Given an $m \times n$ lexicographic grid with half information provided by alarms, two cops can guarantee a win.*

Proof. **PS:** Orient the graph so that the long diagonal edges are horizontal. Place alarms across rows 3 and 4 and continue placing alarms in consecutive rows skipping two rows between. Place any extra alarms across rows closest to $\frac{m}{2}$. We again use consecutive rows because it allows us to know the vertical direction that the robber moves.

SS: This follows directly from Theorem 1 and the proof is identical to that of Theorem 3.2.1. □

Cartesian grids are an interesting case because the results for half information with alarms are exactly the same as those for cameras. This is the case because we were already using two cops on Cartesian grids with cameras and in fact, with full information, two cops are required.

Theorem 3.2.3. *Given an $m \times n$ Cartesian grid with half information provided by alarms, two cops can guarantee a win.*

Proof. **PS:** Orient the graph so $m \leq n$. Place the alarms across every third and fourth row starting with row 3. We place the extra alarms in the empty rows starting with the rows closest to $\frac{m}{2}$.

SS: This follows directly from Theorem 1.

The two cops will search side by side covering two rows at a time.

Case 1: If the robber never lands on an alarm, we search the empty rows until he is apprehended.

Case 2: The robber attempts to get behind us. As soon as he triggers an alarm behind us, we make one move in the vertical direction so we are one row behind him. Now he has two options:

a) If he moves into our row, we continue to move towards him until he enters a different empty row, then we move vertically. This means we are one row behind him moving vertically.

b) If he doesn't move into our row but a row above or below, we move vertically right away.

In the worst case, he only moves vertically and we can chase him to the end of the grid and move closer at each end, eventually catching him. □

Now that we are familiar with alarms on the grids, we generalize our results to $\frac{1}{k}$ information. Before we make any claims about the number of cops required we need to formalize our alarm placement strategy.

Alarm Placement Strategy: Given an $m \times n$ strong, Cartesian or lexicographic grid with $\frac{1}{k}$ information provided by alarms and applying the same placement strategy as used in Theorems 3.2.1, 3.2.2 and 3.2.3, we know that, for this strategy, we need two consecutive rows with alarms. The number of full rows of alarms is given by:

$$2\left\lceil \frac{\min(m,n) - 2k + 1}{2k} \right\rceil + a$$

where

$$a = 1 + \left\lfloor \frac{\min(m,n) - 2k + 1}{2k} \right\rfloor - \left\lceil \frac{\min(m,n) - 2k + 1}{2k} \right\rceil .$$

This means we need a total of

$$\left(2\left\lceil \frac{\min(m,n) - 2k + 1}{2k} \right\rceil + a\right)\max(m,n)$$

$$= \left(2\left\lceil \frac{\min(m,n) + 1}{2k} \right\rceil + a - 2\right)\max(m,n)$$

alarms.

We want to place our consecutive rows of alarms in such a way that we minimize the number of consecutive empty rows. It is difficult to give an exact formula for the number of cops required as was done with cameras because for alarms the cop number is always different for different size graphs. However, we can give upper and lower bounds on the number of cops.

Theorem 3.2.4. *Given an $m \times n$ strong, Cartesian or lexicographic grid with $\frac{1}{k}$ information provided by alarms, the best case for the upper bound on the number of cops is $k-1$ and the worst case for the upper bound on the number of cops is $2(k-1)$.*

Proof. The best case occurs when $\min(m,n) = 0 \pmod{k}$. This is when we have exactly $1, 2, 3...$ full rows of alarms with no extras. If we look at the simplest case, when $\min(m,n) = k$ we know that we will have one full row of alarms. This is because we have $\left\lfloor \frac{k \max(m,n)}{k} \right\rfloor$ alarms available. Simplifying this we get $\max(m,n)$ alarms or one full row of alarms. Placing this full row of alarms in row 1 or row k, we conclude that we need $k-1$ cops.
The worst case occurs when $\min(m,n) = -1 \pmod{2k}$ or just before we get $2, 4, 6...$ full rows of alarms. This means we have many extra alarms with no purpose. Of these cases the worst occurs when $\min(m,n) = 2k-1$ because this is when we have just under 2 full rows of alarms. This means we must place our only full row of alarms in row 1 or $2k-1$ and this means we will need $2k-2$ cops. □

As we can see, for alarms we have a range for the number of cops required depending on the size of the graph and the amount of information given. However, this is still helpful as we are aware of when the best and worst cases occur.

Chapter 4

Full complete k-ary trees

Before we depart from vertex detection devices and proceed to edge detection devices, we will explore cameras and alarms on full complete k-ary trees. This section is different from the rest of this thesis because we are no longer holding the information constant and determining the number of cops. Instead, we are fixing the number of cops at one and determining the least amount of information needed. However, quite a bit of research has already been done on this problem in [2, 3, 4, 6, 7]. That being said, an improvement has been made over [7] on the number of cameras required for one cop in some cases by implementing a new strategy. Although no improvement was made on the number of alarms required for one cop, the results presented give a direct method for calculating the number of alarms required.

4.1 Cameras

We begin by investigating the use of cameras on full complete k-ary trees. Before determining the number of cameras needed for one cop to win, we must introduce the placement strategy. We know that, in general, a cop cannot win on a full complete k-ary tree of depth 2 with no information. However, if we place a camera at the root, the cop can win on a full complete k-ary tree of depth 2. For example, let's consider the full complete ternary tree of depth 2 in Figure 4.1.

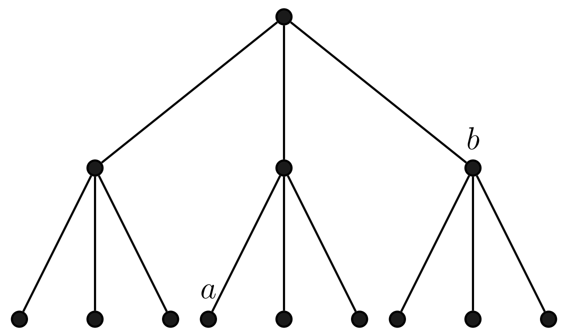

Figure 4.1: Ternary tree of depth 2

Clearly, one cop cannot guarantee a win with no information. This is because the robber can get behind the cop to an area of the tree that has already been searched and this could continue indefinitely. For example, in Figure 4.1, let's assume that the cop has already searched the leftmost section of the tree and is currently occupying vertex a. If the robber is located at vertex b and it is his turn then he can move to the leftmost section without the cop knowing (or being able to stop him). However if we place a camera at the root, then one cop can guarantee a win. Placing the camera at the root allows us to know exactly in which of the three subtrees the robber is located. For trees with depth $d \geq 2$, we place the cameras on the vertices of depth $d-2, d-5, d-8$, etc. Essentially, we are placing the cameras on the root of subtrees of depth 2, starting from the bottom. Let's consider an example and for aesthetic purposes, let's assume that the tree in Figure 4.2 is extended by adjoining a tree like that in Figure 4.1 at each leaf (i.e. the tree in Figure 4.2 now has depth 5). Thus, the first cameras are placed on the depth 3 vertices. More simply, there is a subtree

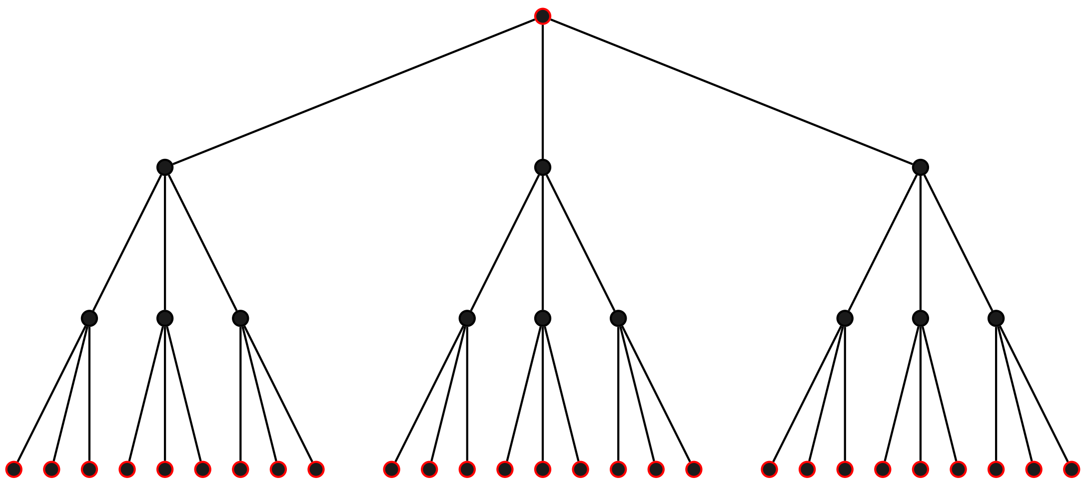

Figure 4.2: Ternary tree of depth 3

of depth 2 identical to that of Figure 4.1 with the root at each depth 3 vertex. The second level of cameras are then placed at depth 0 or the root of the entire tree.

Now that we have outlined the camera placement strategy we can proceed with the following theorem.

Theorem 4.1.1. *Given a full complete k-ary tree of depth d, where $d \geq 2$ and $k \geq 2$, the number of cameras that will suffice for one cop to win is:*

$$k^{(d-2) \pmod 3} \sum_{i=0}^{i=\lfloor \frac{d-2}{3} \rfloor} k^{3i}.$$

Proof. Place the cameras as discussed in the preamble to the theorem. The cop begins searching the tree from the right most leaf. If the robber never triggers a camera, he is apprehended because the vertices without cameras are searchable by one cop. If he does trigger a camera the cop knows his position (and his direction upon leaving) and since there is a unique path between the cop's position and the robber's position, he will eventually force him to a leaf and apprehend him. $\square$

We now want to consider the proportion of cameras to vertices that will suffice for such graphs.

Theorem 4.1.2. *Given a full complete k-ary tree of depth d where $d \geq 2$ and $k \geq 2$, the proportion of cameras (to vertices) that will suffice for one cop to win is:*

$$\frac{1}{k^2 + k + 1}.$$

Proof. First, the total number of vertices on such a tree is given by $\frac{k^{d+1}-1}{k-1}$. Thus the proportion of vertices occupied by cameras is given by:

$$\frac{k^{(d-2) \pmod 3} \sum_{i=0}^{i=\lfloor \frac{d-2}{3} \rfloor} k^{3i}}{\frac{k^{d+1}-1}{k-1}}. \tag{4.1}$$

If we consider what happens as the depth increases to infinity we will be able to make a claim about the proportion of cameras needed on a full complete k-ary tree of any depth. Before attempting to take the limit we should note that:

$$(d-2) \pmod 3 = d - 2 - 3 \left\lfloor \frac{d-2}{3} \right\rfloor.$$

When we take the limit of 4.1 as d approaches infinity we must consider three cases. We should also note that the second derivative of this function is always negative and so the function is concave. This means that the function can never be larger than the limiting value.

Case 1: If $d = 1 \pmod 3$, then we can simplify the floor function as follows:

$$\left\lfloor \frac{d-2}{3} \right\rfloor = \frac{d-4}{3}.$$

Simplifying the numerator of 4.1, we have

$$k^{(d-2) \pmod 3} \sum_{i=0}^{i=\lfloor \frac{d-2}{3} \rfloor} k^{3i}$$

$$= k^{d-2-3\frac{d-4}{3}} \sum_{i=0}^{i=\frac{d-4}{3}} k^{3i}$$

$$= k^2 \sum_{i=0}^{i=\frac{d-4}{3}} k^{3i}$$

$$= k^2 \frac{1-(k^3)^{\frac{d-1}{3}}}{1-k^3}.$$

Putting in the denominator we get

$$k^2 \frac{1-k^{d-1}}{1-k^3} \frac{k-1}{k^{d+1}-1}$$

$$= \frac{k^2(1-k^{d-1})}{(k^2+k+1)(1-k^{d+1})}.$$

Using partial fractions, we can simplify this to

$$\frac{1-k^2}{(k^2+k+1)(-1+k^{d+1})} + \frac{1}{k^2+k+1}.$$

And finally, taking the limit as d approaches infinity, we get

$$\lim_{d\to\infty} \left(\frac{1-k^2}{(k^2+k+1)(-1+k^{d+1})} + \frac{1}{k^2+k+1} \right)$$

$$= \frac{1}{k^2+k+1}.$$

Case 2: If $d = 2 \pmod 3$, then we can simplify the floor function as follows:

$$\left\lfloor \frac{d-2}{3} \right\rfloor = \frac{d-2}{3}.$$

Simplifying the numerator of 4.1 we have

$$k^{(d-2) \pmod 3} \sum_{i=0}^{i=\lfloor \frac{d-2}{3} \rfloor} k^{3i}$$

$$= k^{d-2-3\frac{d-2}{3}} \sum_{i=0}^{i=\frac{d-2}{3}} k^{3i}$$

$$= \sum_{i=0}^{i=\frac{d-2}{3}} k^{3i}$$

$$= \frac{1-(k^3)^{\frac{d+1}{3}}}{1-k^3}.$$

Putting in the denominator we get

$$\frac{1-k^{d+1}}{1-k^3} \frac{k-1}{k^{d+1}-1}$$

$$= \frac{(1-k^{d+1})}{(k^2+k+1)(1-k^{d+1})}.$$

We can simplify this to

$$\frac{1}{k^2+k+1}.$$

Case 3: If $d = 0 \pmod 3$, then we can simplify the floor function as follows:

$$\left\lfloor \frac{d-2}{3} \right\rfloor = \frac{d-3}{3}.$$

Simplifying the numerator of 4.1, we have

$$k^{(d-2) \pmod 3} \sum_{i=0}^{i=\lfloor \frac{d-2}{3} \rfloor} k^{3i}$$

$$= k^{d-2-3\frac{d-3}{3}} \sum_{i=0}^{i=\frac{d-3}{3}} k^{3i}$$

$$= k \sum_{i=0}^{i=\frac{d-3}{3}} k^{3i}$$

$$= k\frac{1-(k^3)^{\frac{d}{3}}}{1-k^3}.$$

Putting in the denominator we get

$$k\frac{1-k^d}{1-k^3}\frac{k-1}{k^{d+1}-1}$$

$$= \frac{k(1-k^d)}{(k^2+k+1)(1-k^{d+1})}.$$

Using partial fractions we can simplify this to

$$\frac{1-k}{(k^2+k+1)(-1+k^{d+1})} + \frac{1}{k^2+k+1}.$$

And finally, taking the limit as d approaches infinity, we get

$$\lim_{d\to\infty}\left(\frac{1-k}{(k^2+k+1)(-1+k^{d+1})} + \frac{1}{k^2+k+1}\right)$$

$$= \frac{1}{k^2+k+1}.$$

$\square$

Now that we have described and proved the results for the new strategy in Theorem 4.1.1, we need to compare it to the best known strategy. The known strategy (call it Strategy 2) is discussed in Section 1.7. This strategy is adapted from photo radar and does not take full advantage of the information provided by cameras. The strategy developed in Theorem 4.1.1 is different in the sense that it only works for cameras and alarms, as we will see in the next section. For simplicity, we will compare the two strategies for ternary trees, T, of increasing depths as shown in Table 4.1. As

Depth of T	V(T)	# of cameras for Strategy 2	# of cameras for Theorem 4.1.1
1	4	0	0
2	13	1	1
3	40	5	3
4	121	15	9
5	364	49	28
6	1093	143	84
7	3280	441	252
8	9841	1303	757
9	29524	3953	2271
10	88573	11775	6813

Table 4.1: Comparing strategies for full complete ternary trees. The results for Strategy 2 are taken from [7].

we can see, the new strategy described in Theorem 4.1.1 does much better than the known strategy as the depth of the tree increases. This holds true for all values of k even though only $k = 3$ (ternary) was shown here.

4.2 Alarms

Alarms are less informative than cameras because they do not show the direction of the robber. Direction is quite valuable in trees because knowing if/when the robber is in a certain branch can be very helpful. Our results here are not an improvement over the method discussed in Section 1.7; however, the results presented here give a simple method for calculating the exact number of alarms required. The placement strategy for alarms is similar to that with cameras except placing an alarm at the root of a depth 2 full, complete k-ary tree will not work as it did with cameras. It won't work because unlike cameras, alarms do not give us direction. In order to replicate the information provided by a camera at the root, we place alarms on the depth 1 vertices of a depth 2 full complete k-ary tree. In Figure 4.3, a depth 2 full, complete ternary tree, we place the alarms on the three depth 1 vertices.

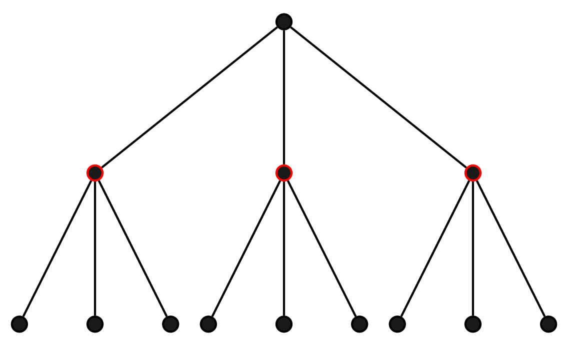

Figure 4.3: Ternary tree of depth 2

From before, we know that in general a full complete k-ary tree of depth 2 cannot be searched with 1 cop and no information. However, if we place a camera at the root, 1 cop can guarantee a win. We can replace this camera at the root with k alarms at the depth one vertices and we end up with equivalent information. As in the case of cameras, placing the alarms on the depth 1 vertices allows us to know in which subtree the robber is located. For trees with depth $d \geq 2$, we place the alarms on the depth $d-1, d-3, d-5$, etc vertices. However, for the case where the root of the tree requires an alarm, (i.e. the root is at depth $d-1, d-3, d-5$, etc) we do not place one. In general, if a vertex requires an alarm, then it must be a depth 1 vertex of a rooted subtree of depth 2. But, because the vertex in question is the root of the entire tree, it cannot have a parent vertex and so we do not have a full complete depth 2 rooted subtree. Thus, no alarm is needed. For example, consider the tree shown in Figure 4.4. It is of depth 3 and so alarms need to be placed on depth 2 vertices and the depth 0 vertex (the root). But, as we just mentioned, we would not place an alarm on the root. Now, consider the tree in Figure 4.4 and let's assume that the tree is of depth 4 (i.e. the leaves are not shown). This means that alarms will be placed on depth 3 and depth 1 vertices.

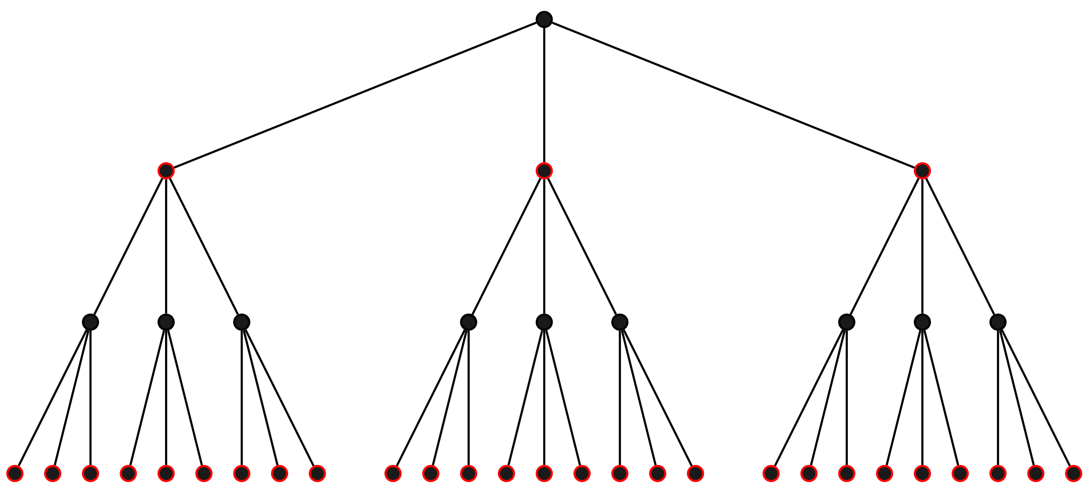

Figure 4.4: Ternary tree of depth 4

Now that we have introduced the placement strategy for alarms, we proceed with the following theorem.

Theorem 4.2.1. *Given a full complete k-ary tree of depth d where $d \geq 2$ and $k \geq 2$, the number of alarms that suffice for one cop to win is:*

$$k^{d \pmod 2} \sum_{i=0}^{i=\lfloor \frac{d}{2} \rfloor - 1} k^{2i+1}.$$

Proof. Place the alarms as discussed in the preamble to the theorem. The cop begins searching the tree from the right most leaf. If the robber never triggers an alarms, he is apprehended because the vertices without alarms are searchable by one cop. If he does trigger an alarm, the cop immediately moves towards the root in order to cut the robber off. This allows the cop to shrink the tree and eventually force the robber to a leaf where he is apprehended. The idea here is that the robber can never get behind the cop. □

We now want to consider the proportion of alarms to vertices that will suffice for such graphs.

Theorem 4.2.2. *Given a full complete k-ary tree of depth d where $d \geq 2$ and $k \geq 2$, the proportion of alarms (to vertices) that will suffice for one cop to win is:*

$$\frac{1}{k+1}$$

Proof. First, the total number of vertices on such a tree is given by $\frac{k^{d+1}-1}{k-1}$. The proportion of vertices occupied by alarms is then given by:

$$\frac{k^{d \pmod 2} \sum_{i=0}^{i=\lfloor \frac{d}{2} \rfloor - 1} k^{2i+1}}{\frac{k^{d+1}-1}{k-1}}. \tag{4.2}$$

If we consider what happens as the depth increases to infinity we can make a claim about the proportion of alarms needed in general. First we should note that

$$d \pmod 2 = d - 2\left\lfloor \frac{d}{2} \right\rfloor.$$

When we take the limit as the depth approaches infinity we must consider two cases. We should also note that the second derivative of this function is always negative and so the function is concave. This means that the function can never be larger than the limiting value.

Case 1: For trees of even depth, $\lfloor \frac{d}{2} \rfloor = \frac{d}{2}$.

Simplifying the numerator in 4.2 we get

$$k^{d \pmod 2} \sum_{i=0}^{i=\lfloor \frac{d}{2} \rfloor - 1} k^{2i+1}$$

$$= k^{d-2\lfloor \frac{d}{2} \rfloor} \sum_{i=0}^{i=\lfloor \frac{d}{2} \rfloor - 1} k^{2i+1}$$

$$= \sum_{i=0}^{i=\frac{d}{2}-1} k^{2i+1}$$

$$= k\frac{1-(k^2)^{\frac{d}{2}}}{1-k^2}.$$

Bringing in the denominator and simplifying, we have

$$k\frac{1-k^d}{1-k^2}\frac{k-1}{k^{d+1}-1}$$

$$= \frac{k(1-k^d)}{(1+k)(1-k^{d+1})}.$$

Using partial fractions we get

$$\frac{1-k}{(k+1)(k^{d+1}-1)}+\frac{1}{k+1}.$$

And finally, taking the limit as d approaches infinity, we get

$$\lim_{d\to\infty}\left(\frac{1-k}{(k+1)(k^{d+1}-1)}+\frac{1}{k+1}\right)$$

$$=\frac{1}{k+1}.$$

Case 2: For trees of odd depth, $\left\lfloor \frac{d}{2} \right\rfloor = \frac{d-1}{2}$.

Simplifying the numerator in 4.2 we get

$$k^{d \pmod 2} \sum_{i=0}^{i=\lfloor \frac{d}{2} \rfloor -1} k^{2i+1}$$

$$= k^{d-2\lfloor \frac{d}{2} \rfloor} \sum_{i=0}^{i=\lfloor \frac{d}{2} \rfloor -1} k^{2i+1}$$

$$= k \sum_{i=0}^{i=\frac{d-1}{2}-1} k^{2i+1}$$

$$= k^2 \frac{1-(k^2)^{\frac{d-1}{2}}}{1-k^2}.$$

Bringing in the denominator and simplifying, we have

$$k^2 \frac{1-k^{d-1}}{1-k^2} \frac{k-1}{k^{d+1}-1}$$

$$= \frac{k^2(1-k^{d-1})}{(1+k)(1-k^{d+1})}.$$

Using partial fractions we get

$$\frac{1-k}{k^{d+1}-1}+\frac{1}{k+1}.$$

And finally, taking the limit as d approaches infinity, we get

$$\lim_{d\to\infty}\left(\frac{1-k}{k^{d+1}-1}+\frac{1}{k+1}\right)$$

$$=\frac{1}{k+1}.$$

$\square$

The results presented here are, in fact, not an improvement of the known results in [2] and [4], which are discussed in Section 1.7. However, these results are quite close and present a simple method for quickly calculating the number of cameras required. The strategy from Section 1.7 does not have a simple method for calculating the number of cameras required, nor a method for calculating the proportion of alarms needed for trees of varying k.

We have concluded the detour into trees and we now proceed to examine photo radar. We will return to the main objective of this thesis, which is to hold the information constant and determine the number of cops required.

Chapter 5

Photo Radar

Photo radar units with and without direction are much different in the information they provide than cameras or alarms. The most important difference is that when we say $\frac{1}{k}$ information for photo radar, we mean $\frac{1}{k}$ of the total edges may have photo radar units placed on them. This is in contrast to alarms and cameras which focus on total vertices. They can however be used in very similar ways and, in fact, Theorem 1 is applied throughout this chapter as our search and placement strategy is unchanged. The main difference between photo radar and cameras is that a robber can stop on any vertex and remain there without being discovered. We will use photo radar in such a way that we know when the robber moves to, and when the robber leaves, a particular vertex. This is in contrast to cameras and alarms which tell us what vertex the robber is occupying. It should be noted that a single camera can give us all the information we need about a certain vertex whereas, depending on the degree of that vertex, we may need many photo radar units to provide the same information. In this chapter, we will examine, in general, grids using photo radar units that provide direction and those that do not.

5.1 With Direction

Photo radar units with direction are used to show us vertical movement similar to cameras. We begin with the Cartesian grid because it has the fewest edges compared to strong and lexicographic grids. Before beginning, we should note that "row" is being used to describe a row of edges. For example, in Figure 5.1, the three thick edges are referred to as a "row" of edges.

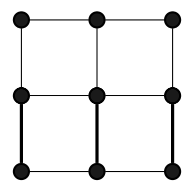

Figure 5.1: 3×3 Cartesian grid

Before we proceed with determining the number of photo radar units needed for a Cartesian grid, we must count the number of edges. The number of edges for an $m \times n$ Cartesian grid is given by: $(m-1)n+(n-1)m = 2mn-n-m$. We should note that in the following theorems we are given $\frac{1}{k}$ information and yet, we are bounding the number of photo radar units. We do this because, depending on the dimensions of the graph under consideration, we may not need all $\frac{1}{k}$ photo radar units. In other words, we include the formula so we can determine exactly how many photo radar units are required for this strategy. This will be done throughout the chapter. Lastly, recall that one cop cannot win on a Cartesian grid with full information.

Theorem 5.1.1. *Given an $m \times n$ Cartesian grid with $\frac{1}{k}$ information, the number of photo radar units that will suffice is given by:*

$$\left\lfloor \frac{m-1}{\lceil \frac{k}{2} \rceil} \right\rfloor n,$$

and the number of cops needed is given by:

$$\left\lceil \frac{k}{2} \right\rceil .$$

Proof. **PS:** Our strategy is to place the photo radar units along vertical edges only. We place the photo radar units every $\lceil \frac{k}{2} \rceil - 1$ "rows" of vertical edges. We must be sure to place them on all vertical edges in a "row" (see the definition of "row" at the beginning of the chapter).

SS: This allows us to search across rows and use the same strategy and argument as for a Cartesian grid with cameras. We will always know when the robber moves vertically and thus we can invoke Theorem 1.

However, we need to prove that we will always have enough photo radar units or equivalently, that:

$$\left\lfloor \frac{m-1}{\lceil \frac{k}{2} \rceil} \right\rfloor n \leq \left\lfloor \frac{2mn-n-m}{k} \right\rfloor .$$

Assume that k is even and note that $2mn - 2n \leq 2mn - n - m$, since $m \leq n$. Let $a = \frac{2m-2}{k}$. Now,

$$\left\lfloor \frac{m-1}{\lceil \frac{k}{2} \rceil} \right\rfloor n = \left\lfloor \frac{2m-2}{k} \right\rfloor n = \lfloor a \rfloor n$$

and

$$\left\lfloor \frac{2mn-n-m}{k} \right\rfloor \geq \left\lfloor \frac{2mn-2n}{k} \right\rfloor = \lfloor na \rfloor .$$

We know that $\lfloor a \rfloor n \leq \lfloor na \rfloor$ from Hermite's Identity.

If k is odd, we have

$$\left\lfloor \frac{m-1}{\lceil \frac{k}{2} \rceil} \right\rfloor n = \left\lfloor \frac{2m-2}{k+1} \right\rfloor n \leq \left\lfloor \frac{2m-2}{k} \right\rfloor n.$$

And so

$$\left\lfloor \frac{m-1}{\lceil \frac{k}{2} \rceil} \right\rfloor n \leq \left\lfloor \frac{2mn-n-m}{k} \right\rfloor.$$

□

As we can see, the number of photo radar units required is much higher than the number of cameras for the same graph. This is always the case and the difference increases as the number of edges in the graph increases.

Strong grids are different in that they are copwin with full information, yet they have many more edges than Cartesian grids. The number of edges in an $m \times n$ strong grid is given by: $(m-1)(5n-4)-(n-1)(m-2) = 4mn-3m-3n+2$. The strategy here for 1 cop on a strong grid differs slightly from the strategy for 2 or more cops on a strong grid. As we will see, 1 cop is sufficient for $\frac{3}{4}$ information. However, if we decrease the amount of information more cops are needed. We begin with the strategy for one cop.

Theorem 5.1.2. *Given an $m \times n$ strong grid with $\frac{3}{4}$ information and 1 cop, $3nm - 3m - 3n + 3$ photo radar units will suffice for the cop to guarantee a win.*

Proof. **PS:** We begin by placing the photo radar units "row" by "row". We place them along each cross (diagonal) edge and all but one vertical edge in each "row".

We leave the last vertical edge empty in "row 1" and the first vertical edge empty in "row 2" and continue alternating in this manner.

SS: This will create one long freepath through the graph for us to search. For example, in Figure 5.2, we have a 4×3 strong grid and the lighter edges indicate the placement of the photo radar units. The remaining thick black edges create a copwin path for us to search.

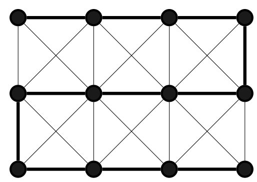

Figure 5.2: 4×3 strong grid

We must prove that $\frac{3}{4}$ information is sufficient. We should also note that the second derivative (with respect to m and n) of this function is always negative and so the function is concave. This means that the function can never be larger than the limiting value. Now,

$$\lim_{\substack{m\to\infty \\ n\to\infty}} \frac{3nm - 3m - 3n + 3}{4nm - 3m - 3n + 2}.$$

Using L'Hopital's rule with respect to n, we get

$$= \lim_{\substack{m\to\infty \\ n\to\infty}} \frac{3m - 3}{4m - 3}.$$

And using L'Hospital's rule with respect to m, we have

$$\lim_{\substack{m\to\infty \\ n\to\infty}} \frac{3m - 3}{4m - 3} = \frac{3}{4}.$$

□

Now we proceed to our strategy for more than two cops. In the next theorem, we begin with $\frac{3}{4k}$ information and determine that we need a minimum of k cops. However, given those k cops, we do not need the full $\frac{3}{4k}$ information and so we give a bound for the number of photo radar units. We generalize our results with the following theorem.

Theorem 5.1.3. *Given an $m \times n$ strong grid with $\frac{3}{4k}$ information and $k \geq 2$ cops, the number of photo radar units that will suffice is given by*

$$\left\lfloor \frac{m-1}{k} \right\rfloor (3(n-1)+1).$$

Proof. **PS:** Our strategy is to place photo radar units on all vertical and cross edges in a given "row". The $\left\lfloor \frac{m-1}{k} \right\rfloor$ tells us how many rows we must cover and we want to minimize the number of consecutive empty "rows" between them.
SS: Our search strategy is the same as mentioned earlier for cameras and thus Theorem 1 applies. However, we must prove that our information required is correct.

The proof here and those that follow will make use of the squeeze theorem in order to deal with the floor functions. First, we know that

$$\frac{m-1}{k} - 1 \leq \left\lfloor \frac{m-1}{k} \right\rfloor \leq \frac{m-1}{k}.$$

We begin with the upper bound. Again, we should note that the second derivative of this function is always negative and so the function is concave. This means that the function can never be larger than the limiting value. Now,

$$\lim_{\substack{m\to\infty \\ n\to\infty}} \frac{\frac{m-1}{k}(3(n-1)+1)}{4nm-3n-3m+2}$$

$$= \lim_{\substack{m\to\infty \\ n\to\infty}} \frac{(m-1)(3n-2)}{4nmk - 3nk - 3mk + 2k}$$

$$= \lim_{\substack{m\to\infty \\ n\to\infty}} \frac{3nm - 2m - 3n + 2}{4nmk - 3nk - 3mk + 2k}.$$

Using L'Hopital's rule with respect to n, we get

$$\lim_{\substack{m\to\infty \\ n\to\infty}} \frac{3m-3}{4mk-3k}.$$

And with respect to m, we get

$$\frac{3}{4k}.$$

Now, for the lower bound we have,

$$\lim_{\substack{m\to\infty \\ n\to\infty}} \frac{\frac{m-1-k}{k}(3(n-1)+1)}{4nm - 3n - 3m + 2}$$

and so

$$\lim_{\substack{m\to\infty \\ n\to\infty}} \frac{3nm - 3n - 3nk - 2m + 2 + 2k}{4nmk - 3nk - 3mk + 2k}.$$

Again, using L'Hopital's rule with respect to n and then with respect to m, we have $\frac{3}{4k}$. And since the upper and lower bound are equal we can invoke the squeeze theorem and conclude that

$$\lim_{\substack{m\to\infty \\ n\to\infty}} \frac{\left\lfloor \frac{m-1}{k} \right\rfloor (3(n-1)+1)}{4nm - 3n - 3m + 2} = \frac{3}{4k}.$$

$\square$

The lexicographic grid is the most difficult grid with photo radar because of the large number of edges. This is in contrast to cameras and alarms where the lexicographic grid was the easiest. The number of edges in an $m \times n$ lexicographic is given by:

$$(m-1)(n^2+2n-2)-(n-1)(m-2)=n^2m-n^2+nm-m.$$

The strategy and proof for lexicographic is essentially the same as the strategy for 2 or more cops on strong grids.

Theorem 5.1.4. *Given an $m \times n$ lexicographic grid with $\frac{1}{k}$ information and k cops, the number of photo radar units that will suffice is given by*

$$\left\lfloor \frac{m-1}{k} \right\rfloor n^2.$$

Proof. **PS:** Our strategy is to cover all vertical and cross edges in "rows", similar to the strategy used for strong grids.

SS: This placement allows us to know whenever the robber moves vertically (between freepaths) and thus we can invoke Theorem 1. However, we must first prove that the information required is satisfied. Recall that,

$$\frac{m-1}{k}-1 \leq \left\lfloor \frac{m-1}{k} \right\rfloor \leq \frac{m-1}{k}.$$

Again, this proof will make use of the squeeze theorem and we begin with the upper bound. We should again note that the second derivative of this function is always negative and so the function is concave. This means that the function can never be larger than the limiting value. Now,

$$\lim_{\substack{m\to\infty \\ n\to\infty}} \frac{\frac{m-1}{k}n^2}{n^2m-n^2+nm-m}$$

$$= \lim_{\substack{m\to\infty \\ n\to\infty}} \frac{mn^2 - n^2}{kn^2m - kn^2 + nmk - mk}.$$

Using L'Hopital's rule twice with respect to n, we get

$$\lim_{\substack{m\to\infty \\ n\to\infty}} \frac{2m-2}{2km-2k}.$$

And using L'Hopital's rule again with respect to m, we get

$$\frac{2}{2k} = \frac{1}{k}.$$

Now for the lower bound, we have

$$\lim_{\substack{m\to\infty \\ n\to\infty}} \frac{(m-1-k)n^2}{kn^2m - kn^2 + nmk - mk}.$$

Using L'Hopitals rule with respect to m, we get

$$= \lim_{\substack{m\to\infty \\ n\to\infty}} \frac{n^2}{kn^2 + nk - k}.$$

And using L'Hopital's rule twice with respect to n, we get

$$\frac{2}{2k} = \frac{1}{k}.$$

We can see that both bounds are equal once again and invoking the squeeze theorem allows us to conclude that

$$\lim_{\substack{m\to\infty \\ n\to\infty}} \frac{\left\lfloor \frac{m-1}{k} \right\rfloor n^2}{n^2m - n^2 + nm - m} = \frac{1}{k}.$$

$\square$

We have now generalized results for all three grids using photo radar units with direction. Although our strategy is not an improvement from the strategy discussed in Section 1.7 and found in [6], it does allow us to make claims on the number of photo radar units needed for more than 2 cops. Strategy 3 is an adaptation of the strategy discussed in Section 1.7 and can be found in [7]. Again, although we do not improve upon this strategy, we do give results for $k > 2$. A comparison between the strategies developed in Theorems 5.1.1, 5.1.3, and 5.1.4 and Strategy 3 can be found in Tables 5.1, 5.2, and 5.3. As we can see, both strategies give the same number of photo radar units and this is because they both use the idea of freepaths or equivalently, copwin bands. Strategy 3 does not give an explicit relationship for calculating the number of photo radar units and determining the number of photo radar units is tedious. However, our method has simple equations for determining the number of vertices and photo radar units needed, which can be easily programmed into a computer for quick results.

Cartesian grid with 2 cops				
m	n	E(G)	# of photo radar for Theorem 5.1.1	# of photo radar for Strategy 3
4	5	31	15	15
5	5	40	20	20
6	6	60	30	30
8	8	112	56	56

Table 5.1: Comparing strategies for Cartesian grids

Strong grid with 2 cops				
m	n	E(G)	# of photo radar for Theorem 5.1.3	# of photo radar for Strategy 3
4	5	55	13	13
5	5	72	26	26
6	6	110	32	32
6	7	131	38	38
8	9	239	75	75

Table 5.2: Comparing strategies for strong grids

lexicographic grid with 2 cops				
m	n	E(G)	# of photo radar for Theorem 5.1.4	# of photo radar for Strategy 3
4	5	91	25	25
5	5	120	50	50
6	6	210	72	72
6	7	281	98	98
8	9	631	243	243

Table 5.3: Comparing strategies for lexicographic grid

5.2 Without Direction

Photo radar units without direction are similar to photo radar units with direction when two are placed adjacent to each other. We will use a strategy very similar to that of alarms: having two adjacent "rows" of photo radar units (recall that "row" is being used to describe a row of edges). We want to minimize the number of adjacent rows without photo radar units in order to minimize the number of cops needed. This strategy will be used throughout this section. This section is also different in the sense that we began as before by fixing the amount of information and determining the number of cops required. However, we then realized that we had some extra information (as before, see Chapter 2) and so we decided to bound the information as well as the cops. First, we determine the number of “rows” that need to be covered using this strategy. We must always have two adjacent “rows” of photo radar units, unless the photo radar units are placed on the outer edges of the graph. We can generalize the number of complete “rows” required for an $m \times n$ grid and it is given by:

$$2\left\lceil\frac{m-k}{k+1}\right\rceil+\left\lceil\frac{m}{k+1}\right\rceil-\left\lceil\frac{m-k}{k+1}\right\rceil-1.$$

Converting the ceilings to floors, we get an upper bound for this expression which will be used later. It is given by

$$2\left\lfloor\frac{m-k-1}{k+1}\right\rfloor+2.$$

We again begin with the simplest case, a Cartesian grid. The number of edges in an $m \times n$ Cartesian grid is given by $(m-1)n+(n-1)m = 2mn-n-m$. Recall again that one cop cannot win on a Cartesian grid with full information.

Theorem 5.2.1. *Given an $m \times n$ Cartesian grid with $\frac{3}{2k+2}$ information and k cops, the number of photo radar units that will suffice is given by*

$$\left(2\left\lfloor\frac{m-k-1}{k+1}\right\rfloor+2\right)n+\left\lfloor\frac{2\left\lfloor\frac{m-k-1}{k+1}\right\rfloor+2}{2}\right\rfloor(n-1).$$

Or equivalently,

$$\left(2\left\lfloor\frac{m-k-1}{k+1}\right\rfloor+2\right)n+\left(\left\lfloor\frac{m-k-1}{k+1}\right\rfloor+1\right)(n-1).$$

Proof. **PS:** Orient the graph so $m \leq n$. Our strategy is to place the photo radar units along vertical edges only. We must be sure to place them on all vertical edges in a “row” and we must ensure that two adjacent “rows” are covered. We also want to minimize the number of empty rows between and thus minimize the number of cops required.

SS: Our placement allows us to search across rows and use the same strategy and argument as Cartesian grid with alarms. We will always know when the robber moves vertically (between freepaths) and thus we can invoke Theorem 1. However, we must first show that our required information is satisfied. We will again be making use of the squeeze theorem and we know that

$$\frac{m-k-1}{k+1} - 1 \leq \left\lfloor \frac{m-k-1}{k+1} \right\rfloor \leq \frac{m-k-1}{k+1}.$$

We begin with the upper bound. Now,

$$\lim_{\substack{m\to\infty \\ n\to\infty}} \frac{\left(2\left(\frac{m-k-1}{k+1}\right)+2\right)n + \left(\left(\frac{m-k-1}{k+1}\right)+1\right)(n-1)}{2mn-n-m}$$

$$= \lim_{\substack{m\to\infty \\ n\to\infty}} \frac{\frac{2mn}{k+1} + \frac{mn-m}{k+1}}{2mn-n-m}$$

$$= \lim_{\substack{m\to\infty \\ n\to\infty}} \frac{3mn-m}{2nmk-nk-mk+2mn-n-m}.$$

Using L'Hopital's rule with respect to n, we get

$$\lim_{\substack{m\to\infty \\ n\to\infty}} \frac{3m}{2mk-k+2m-1}.$$

And using L'Hopital's rule with respect to m, we get

$$\frac{3}{2k+2}.$$

Now for the lower bound, we have

$$\lim_{\substack{m\to\infty \\ n\to\infty}} \frac{\left(2\left(\frac{m-k-1}{k+1}-1\right)+2\right)n+\left(\left(\frac{m-k-1}{k+1}-1\right)+1\right)(n-1)}{2mn-n-m}$$

$$= \lim_{\substack{m\to\infty \\ n\to\infty}} \frac{3n-3kn-3n-m+k+1}{(k+1)(2mn-n-m)}.$$

Using L'Hopital's rule with respect to n, we get

$$\lim_{\substack{m\to\infty \\ n\to\infty}} \frac{3m-3k-3}{2mk-k+2m-1}.$$

And using L'Hopital's rule with respect to m, we get

$$\frac{3}{2k+2}.$$

We can see that both the upper and lower bound are equal and thus invoking the squeeze theorem we can conclude that

$$\lim_{\substack{m\to\infty \\ n\to\infty}} \frac{\left(2\left\lfloor\frac{m-k-1}{k+1}\right\rfloor+2\right)n+\left(\left\lfloor\frac{m-k-1}{k+1}\right\rfloor+1\right)(n-1)}{2mn-n-m} = \frac{3}{2k+2}.$$

□

Proceeding to strong grids, the number of edges for an $m \times n$ strong grid is given by $(m-1)(5n-4)-(n-1)(m-2) = 4mn-3m-3n+2$.

Theorem 5.2.2. *Given an $m \times n$ strong grid with $\frac{7}{4k+4}$ information and k cops, the number of photo radar units that will suffice is given by*

$$\left(2\left\lfloor\frac{m-k-1}{k+1}\right\rfloor+2\right)(3n-2)+\left\lfloor\frac{2\left\lfloor\frac{m-k-1}{k+1}\right\rfloor+2}{2}\right\rfloor(n-1).$$

Or equivalently,

$$\left(2\left\lfloor\frac{m-k-1}{k+1}\right\rfloor+2\right)(3n-2)+\left(\left\lfloor\frac{m-k-1}{k+1}\right\rfloor+1\right)(n-1).$$

Proof. **PS:** Orient the graph so $m \leq n$. Again, we will place the photo radar units in consecutive "rows", similar to the strategy used in Theorem 5.2.1.

SS: Our specific placement of devices allows us to know when the robber moves vertically and thus invoke Theorem 1. First, however, we must show that our required information is satisfied. We begin with the upper bound. Now,

$$\lim_{\substack{m\to\infty\\ n\to\infty}} \frac{\left(2\left(\frac{m-k-1}{k+1}\right)+2\right)(3n-2)+\left(\left(\frac{m-k-1}{k+1}\right)+1\right)(n-1)}{4mn-3m-3n+2}$$

$$= \lim_{\substack{m\to\infty\\ n\to\infty}} \frac{2m(3n-2)+m(n-1)}{(k+1)(4mn-3m-3n+2)}$$

$$= \lim_{\substack{m\to\infty\\ n\to\infty}} \frac{6mn-4m+mn-m}{4mnk-3mk-3nk+2k+4mn-3m-3n+2}.$$

Using L'Hopital's rule with respect to n, we get

$$\lim_{\substack{m\to\infty\\ n\to\infty}} \frac{6m+m}{4mk-3k+4m-3}.$$

And again using L'Hopital's rule with respect to m, we get

$$\frac{7}{4k+4}.$$

Now for the lower bound, we have

$$\lim_{\substack{m\to\infty\\ n\to\infty}} \frac{\left(2\left(\frac{m-k-1}{k+1}-1\right)+2\right)(3n-2)+\left(\left(\frac{m-k-1}{k+1}-1\right)+1\right)(n-1)}{4mn-3m-3n+2}$$

$$= \lim_{\substack{m\to\infty \\ n\to\infty}} \frac{7mn - 7nk - 7n - 5m + 5k + 5}{4mnk - 3mk - 3nk + 2k + 4mn - 3m - 3n + 2}.$$

Using L'Hopital's rule with respect to n, we get

$$\lim_{\substack{m\to\infty \\ n\to\infty}} \frac{7m - 7k - 7}{4mk - 3k + 4m - 3}.$$

And using L'Hopital's rule with respect to m, we get

$$\frac{7}{4k+4}.$$

Since both the upper and lower bounds are equal, we can invoke the squeeze theorem and conclude that

$$\lim_{\substack{m\to\infty \\ n\to\infty}} \frac{\left(2\left\lfloor \frac{m-k-1}{k+1} \right\rfloor + 2\right)(3n-2) + \left(\left\lfloor \frac{m-k-1}{k+1} \right\rfloor + 1\right)(n-1)}{4mn - 3m - 3n + 2} = \frac{7}{4k+4}.$$

$\square$

Lastly, we consider lexicographic grids. The number of edges in an $m \times n$ lexicographic grid is given by $(m-1)(n^2+2n-2) - (n-1)(m-2) = n^2m - n^2 + nm - m$.

Theorem 5.2.3. *Given an $m \times n$ lexicographic grid with $\frac{2}{k+1}$ information and k cops, the number of photo radar units that will suffice is given by*

$$\left(2\left\lfloor \frac{m-k-1}{k+1} \right\rfloor + 2\right) n^2 + \left\lfloor \frac{2\left\lfloor \frac{m-k-1}{k+1} \right\rfloor + 2}{2} \right\rfloor (n-1).$$

Or equivalently,

$$\left(2\left\lfloor \frac{m-k-1}{k+1} \right\rfloor + 2\right) n^2 + \left(\left\lfloor \frac{m-k-1}{k+1} \right\rfloor + 1\right)(n-1).$$

Proof. **PS:** Orient the graph so $m \leq n$. Again, we will place the photo radar units in consecutive "rows", similar to the strategy used for Theorems 5.2.1 and 5.2.2.

SS: This allows us to know when the robber moves vertically and thus invoke Theorem 1. First we must show that our required information is satisfied. We again begin with the upper bound. Now

$$\lim_{\substack{m\to\infty \\ n\to\infty}} \frac{\left(2\left(\frac{m-k-1}{k+1}\right)+2\right)n^2 + \left(\left(\frac{m-k-1}{k+1}\right)+1\right)(n-1)}{n^2m - n^2 + nm - m}$$

$$= \lim_{\substack{m\to\infty \\ n\to\infty}} \frac{2mn^2 + mn - m}{(k+1)(n^2m - n^2 + nm - m)}.$$

Using L'Hopital's rule twice with respect to n, we get

$$\lim_{\substack{m\to\infty \\ n\to\infty}} \frac{4m}{2mk - 2k + 2m - 2}.$$

And using L'Hopital's rule with respect to m, we get

$$\frac{2}{k+1}.$$

And for the lower bound, we have

$$\lim_{\substack{m\to\infty \\ n\to\infty}} \frac{\left(2\left(\frac{m-k-1}{k+1}-1\right)+2\right)n^2 + \left(\left(\frac{m-k-1}{k+1}-1\right)+1\right)(n-1)}{n^2m - n^2 + nm - m}.$$

Applying L'Hopital's rule with respect to m, we get

$$\lim_{\substack{m\to\infty \\ n\to\infty}} \frac{2n^2 + n - 1}{n^2k + nk - k + n^2 + n - 1}.$$

And applying L'Hopital's rule with respect to n twice, we get

$$\frac{2}{k+1}.$$

As we can see, both bounds are equal and thus, we conclude using the squeeze theorem that

$$\lim_{\substack{m\to\infty\\ n\to\infty}} \frac{\left(2\left\lfloor \frac{m-k-1}{k+1}\right\rfloor + 2\right)n^2 + \left(\left\lfloor \frac{m-k-1}{k+1}\right\rfloor + 1\right)(n-1)}{n^2m - n^2 + nm - m} = \frac{2}{k+1}.$$

$\square$

The results presented above are in fact a slight improvement on the best current strategy for photo radar units without direction on these classes of graphs, which is discussed in Section 1.7 and can be found in [7]. The best current strategy (call it Strategy 3) is generalized for all copwin graphs and does not take advantage of the unique characteristics of grids. This allowed improvements to be made and the comparison between methods is shown below in Tables 5.4, 5.5, and 5.6. Strategy 3 does not give explicit formulae for calculating the number of photo radar units, which can be difficult. We have direct methods for calculating the number of photo radar units and edges in each class of graphs, which can be easily programmed for quick results. We should note that the number of photo radar units needed for 4×5 and 5×5 grids is the same. The reason for this is that both have $n = 5$ and the floor function for both rounds down to zero, yielding the same result.

Cartesian grid with 2 cops				
m	n	E(G)	# of photo radar for Theorem 5.2.1	# of photo radar for Strategy 3
4	5	31	14	17
5	5	40	14	20
6	6	60	34	34
8	8	112	46	60

Table 5.4: Comparing strategies for Cartesian grids

Strong grid with 2 cops				
m	n	E(G)	# of photo radar for Theorem 5.2.2	# of photo radar for Strategy 3
4	5	55	30	40
5	5	72	30	50
6	6	110	74	80
6	7	131	88	95
8	9	239	116	175

Table 5.5: Comparing strategies for strong grids

lexicographic grid with 2 cops				
m	n	E(G)	# of photo radar for Theorem 5.2.3	# of photo radar for Strategy 3
4	5	91	54	70
5	5	120	54	100
6	6	210	154	180
6	7	281	208	240
8	9	631	340	560

Table 5.6: Comparing strategies for lexicographic grid

Chapter 6

Conclusions & Further Research

6.1 Conclusions

In this thesis, we look at the Cops and Robber game with partial information. The main focus is to fix the amount of information provided and determine the number of cops required to apprehend the robber. The main graphs we investigate are lexicographic, strong, and Cartesian grids. We first develop bounds on the number of cops required for the three grids, given $\frac{1}{k}$ information provided by cameras. Some simple extensions are made to bound the number of cops required on n-dimensional Cartesian grids and bipartite graphs. Next, we bound the number of cops required given $\frac{1}{k}$ information provided by alarms on the three grids. A diversion is made in Chapter 4, where we fix the number of cops at one and determine the amount of information needed on full complete k-ary trees. The results found using cameras are an improvement on the known bounds for one cop. The results for alarms using the same strategy as cameras are presented, although they are not an improvement on the known bounds for one and two cops. Finally, a relationship is developed between number of cops and cameras required for photo radar units on the three grids. Bounds for the number of cops required using photo radar units with direction on these grids are presented. The bounds for photo radar units without direction are shown to be an improvement on the known bounds and comparisons are given.

6.2 Further Research

This thesis and the results presented lead to many intriguing problems that are open for further work. The main problem of generalizing schemes for placing information devices on all copwin graphs when k cops are playing the game is still open. However, another area for further exploration is the problem of k cops playing with partial information on k-copwin graphs. The characterization for k-copwin graphs can be found in [5].

Another area for further research is the problem of fixing the information and determining bounds on the number of cops. The only classes of graphs considered for this problem are those presented in this thesis.

Lastly, this thesis leaves the reader with the problem of improving the bounds already known. Since most of the known bounds are generalized for all copwin graphs, they can be improved for specific classes or types of graphs.

Bibliography

[1] B. Alspach, *Searching & Sweeping Graphs: A Brief Survey*, Le Matematiche **59** (2004), 5-37.

[2] N.E. Clarke, *A game of Cops and Robber played with partial information*, Congressus Numerantium **166** (2004), 145-149.

[3] N.E. Clarke, *Constrained Cops and Robber*, Doctoral Thesis, Dalhousie University, 2002.

[4] N.E. Clarke and E.L. Connon, *Cops, Robber, and Alarms*, Ars Combinatoria **81** (2006), 283-296.

[5] N.E. Clarke and G. MacGillivray, *Characterizations of k-copwin graphs*, Discrete Mathematics **312** (2012), 1421-1425.

[6] N.E. Clarke and R.J. Nowakowski, *Cops, Robber, and Photo Radar*, Ars Combinatoria **56** (2000), 97-103.

[7] D. Jeliazkova, *Aspects of the Cops and Robber Game Played with Incomplete Information*, Masters Thesis, Acadia University, 2006.

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

[9] A. Quilliot, *Thèse d'Etat*, Université de Paris IV, 1983.

[10] D.B. West, *Introduction to Graph Theory*, Prentice-Hall Inc., NJ, 2001.