

COPS, ROBBERS, AND BARRICADES

by

Erin Kathleen McKenna Meger

A thesis presented to Ryerson University
in partial fulfillment of the
requirements for the degree of
Master of Science
in the Program of
Applied Mathematics

Toronto, Ontario, Canada, 2016

© Erin Meger 2016

AUTHOR'S DECLARATION FOR ELECTRONIC SUBMISSION OF
A THESIS

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

I understand that my thesis may be made electronically available to the public.

Abstract

Cops, Robbers, and Barricades is a new variant of the classic game on graphs, Cops and Robbers. In this variant, the robber is granted the ability to build a barricade on an adjacent vertex during her turn. This barricade restricts the cops and the robber from entering this vertex. Analogously to the original game, the minimum number of cops required to guarantee that the cops are able to capture the robber in a finite number of rounds when playing on a graph G is called the *barricade-cop number*, denoted $c_B(G)$. If $c_B(G) = 1$, then G is called *barricade-cop-win*. The game can be generalized so that the robber may build $b(k)$ -many barricades on vertices during her k^{th} turn, in accordance with *barricade rules* that dictate the permissible positions of these barricades.

At the beginning of this thesis, the new variant of the game will be explored on various classes of graphs. This graph parameter is determined exactly for complete graphs, cycles, and paths, but only upper bounds are known for trees and locally-path-like graphs.

Subsequently, three variations to the barricade rules are studied: placing a barricade anywhere on the graph; placing a barricade on neighbouring vertices; and placing a barricade only after moving off of a vertex. The relation between these three variations will be explored.

In the final chapter of this thesis, an algorithmic characterization of barricade-cop-win graphs is given when playing with $b(k)$ -many barricades and any barricade rules. The thesis concludes with a discussion on the many open problems inspired by this new variant of the game of Cops and Robbers.

Acknowledgements

I would like to thank my family first and foremost for their continued support throughout my mathematics education, and for encouraging me to believe in the power of my dreams.

I am thankful for the unending support from my supervisor Dr. Anthony Bonato, for his constant support throughout my Masters degree.

My appreciation goes to Dr. Dejan Delić and Dr. Paweł Prałat for their time and energy in participating in my thesis committee. I also thank Dr. Peter Danziger, whose small side project I never truly got a chance to work on.

I acknowledge the Queen Elizabeth II Graduate Scholarship for Science and Technology and the Ryerson Graduate Fellowship for partially funding this thesis.

Contents

List of Figures	viii
Chapter 1. Introduction	1
1. Motivation	1
2. Introduction to Graph Theory	3
3. Graph Parameters and Domination	10
4. Introduction to Cops and Robbers	12
5. A Brief Discussion on Algorithms	14
6. Outline of Thesis	16
Chapter 2. Cops, Robbers, and Barricades	18
1. An Introduction to the Game	18
2. Preliminary Results	20
3. Trees	26
4. Complete m -ary Trees	27
5. Locally Path-Like Graphs	33
6. Meyniel's Conjecture and Domination	35
Chapter 3. Variations of the Barricade Rules	40
1. Introduction	40

2. Three Variants of the Barricade Rules	40
3. Relating the Three Variants	45
Chapter 4. Characterizing Barricade-Cop-Win Graphs	51
1. A relation on cop-win graphs	53
2. Characterization	54
3. Algorithm	56
4. Complexity	58
Chapter 5. Conclusion and Open Problems	59
1. Conclusion	59
2. Open Problems	60
Appendix A - Code	62
3. m-ary tree bound	62
Bibliography	66

List of Figures

1	An example of Cops and Robbers: rounds 1 and 2.	2
2	An example of Cops and Robbers: rounds 3 and 4.	2
3	On the left, a multi-graph. On the right, a simple graph.	4
4	An example of a subgraph, an induced subgraph, and a spanning subgraph.	5
5	A disconnected graph.	6
6	The graph K_n for $n = 1, 2, 3, 4$.	6
7	Independent set on 4 vertices.	7
8	The graph P_6 .	7
9	The graph C_6 .	7
10	On the left a bipartite graph. On the right $K_{3,3}$.	8
11	Examples of trees.	9
12	The Petersen graph.	11
1	Example: Cops, Robbers and Barricades on P_8 .	21
2	Example 2. The cops play on either end of the path.	22
3	CRB on P_8 where the cops are placed optimally and win.	23

4	A cycle with a sample of the cop placement for CRB.	24
5	An example of a robber-star.	25
6	Complete 3-ary tree of height 2. The height of each vertex is labeled.	27
7	A row in a complete m-ary tree.	29
8	An example of a locally path like graph G , and its corresponding weighted multi-graph $W(G)$.	34
9	The initial set up. The blue edges may exist but are of no consequence to the proof, and the red edges cannot exist due to the girth of the graph being greater than or equal to 7.	37
10	The first move of the cops, before the robber moves.	38
1	Example of the global variant.	41
2	Example of the slug variant of CRB.	43
3	The second-to-last move in the slug version example.	44
4	An example of $N_6(G_1, \dots, G_i, k)$.	47
5	A particular necklace graph with eight pearls.	47
6	The initial positions for the local game of CRB on this graph.	48
7	Game play of the local version of CRB on this particular section of a necklace graph.	48

8	Game play of the global version of CRB, on this particular section of a necklace graph.	50
1	An example of CRB played on a 5×5 strong product grid with $b(k) = 2$ with a vertex cut of size four.	52
2	$G_k = G \langle V(G) \setminus B_k \rangle$ for the example played in Figure 31.	53

CHAPTER 1

Introduction

1. Motivation

In 1978 Quilliot introduced in his doctoral thesis a game that would pioneer a new area of graph theory [19]. When Nowakowski and Winkler independently defined this game it catapulted the topic into the main stream of graph theory [17]. This game is called Cops and Robbers, and it has been of increasing interest to graph theorists since its inception. Researchers have studied the original game, investigating the cop number of a graph, as well as many variants of the original game, such as tandem-win Cops and Robbers, distance k Cops and Robbers, and Zombies and Survivors; see [1, 4, 5, 6, 8, 11, 14].

The game of *Cops and Robbers* is a vertex-pursuit game played on a graph. For now, we can think of a graph as a game board. The game is played as follows. We have two players, the *robber* and the *cops*. It is permissible for the cop player to have a set of cops at their disposal. First, the cops are placed on the “board”, which are simply vertices of the graph. Then the robber places herself on another vertex. The two players alternate turns. If we have more than one cop, then the cop player moves all the cops in each turn. In any given round, either player

may “pass” by remaining on their vertex. The cop wins if he captures the robber, and the robber wins if she can avoid capture indefinitely. In order to know if this is possible, we will need to know much about the structure of the game board; that is, the graph on which the game is played. We will provide a more thorough explanation of the game in the last section of this chapter. For now, let us provide an example of the game.

We begin with a graph below, playing with a single cop and a single robber. The cop places himself on the left most vertex first, and the robber will place herself as far away from the cop as possible.

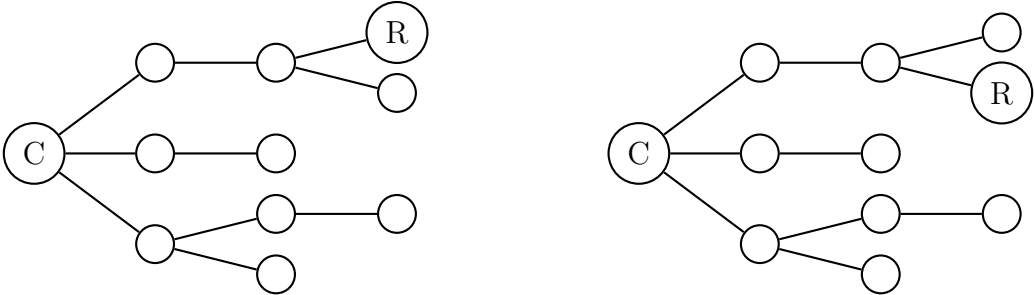


FIGURE 1. An example of Cops and Robbers: rounds 1 and 2.

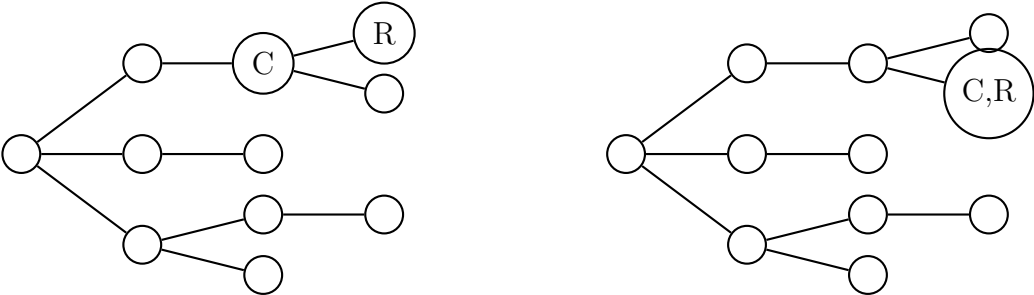


FIGURE 2. An example of Cops and Robbers: rounds 3 and 4.

In the second round, the cop moves first and he moves closer to the robber. The robber has only one place she can go, but this will bring her closer to the cop. So she will stay put and await capture.

2. Introduction to Graph Theory

2.1. Basic Graph Theory Terminology. We define basic graph theoretic notation, and summarize basic results in graph theory used in the thesis. A *graph* is a pair $G = (V, E)$, where V is the non-empty set of *vertices* $\{v_0, v_1, \dots, v_n\}$, and E is the set of *edges* $\{e_0, e_1, \dots, e_m\}$, where each edge is a pair of vertices $e_k = \{v_i, v_j\}$ with $i \neq j$. We can represent vertices, often called *nodes*, by dots and we draw an edge between vertices v_i , and v_j if the edge $\{v_i, v_j\}$ is in $E(G)$. To denote edges we will use the notation $v_i v_j$. It is possible to have graphs with an infinite number of vertices; however, for the remainder of this thesis we will only consider finite graphs unless otherwise stated. Additionally, we make the notational convention: \mathbb{N} represents the natural numbers including zero, and \mathbb{N}^+ is the natural numbers excluding zero.

When $\{v_i, v_j\} \in E(G)$, we say that v_i is *adjacent* to v_j , or that they are *neighbours*. We can denote this $v_i \in N(v_j)$, where $N(v_j)$ is the *neighbour-set* of v_j . Additionally, we use the notation $N[v_j]$ to denote the set of $N(v_j) \cup \{v_j\}$, called the *closed neighbour set*. We say that an edge is *incident* to a vertex, that is e_k is incident to v_i . We define the *degree* of a vertex, written $\deg(v_i)$, to be the number of edges incident

to v_i . We denote the *minimum degree* of all vertices in G as $\delta(G)$, and we denote the *maximum degree* of all vertices in G as $\Delta(G)$. The k^{th} -*neighbour set* of a vertex u , denoted $N_k(u)$, as the set of vertices v such that the minimum distance between u and v is at most k .

In Figure 2 above, we notice that we have two edges from the left vertex to the top vertex (and from the left to the bottom as well). When we have multiple edges, we call this graph a *multi-graph*. In a multi-graph, we can also have a loop on a vertex. Typically, we consider only graphs with single edges between pairs of vertices, with no loops. This kind of graph is called a *simple graph*. From now on, assume we are discussing simple graphs when we say the word “graph” unless otherwise specified.



FIGURE 3. On the left, a multi-graph. On the right, a simple graph.

A *subgraph* $H \subseteq G$ is the graph such that $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$. That is, we take some vertices of G , and only choose some edges between those vertices. An *induced subgraph* H of G that is induced by

the set $U \subseteq V(G)$, denoted $G\langle U \rangle$, is the graph such that $V(H) = U$ and $\{u, v\} \in E(H)$ if and only if $u \in U$ and $v \in U$. That is, H is the graph where we consider vertices in the set U , and include all edges where both endpoints are in U . We call a subgraph with $V(H) = V(G)$ a *spanning subgraph*.

In the example below we have a graph, G , on the left. The graph in the middle is a subgraph of G but is not an induced subgraph since it is missing the edge $\{1, 2\}$. The graph on the right is an induced subgraph using the set $U = \{1, 2, 3\}$.

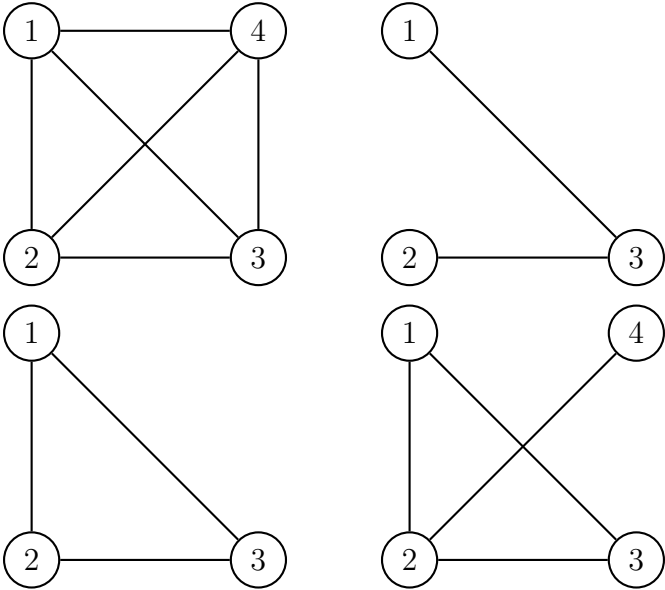


FIGURE 4. An example of a subgraph, an induced subgraph, and a spanning subgraph.

A graph G is called *connected* if and only if for every pair of vertices $u, v \in V(G)$, there exists a path from u to v , that is there is a list of edges we can follow to get from u to v . It is possible that a graph contains an isolated vertex, that is a vertex with no incident edges (that is a vertex

with degree 0). Up until this point, we have not explored graphs that are not connected. Below is an example.

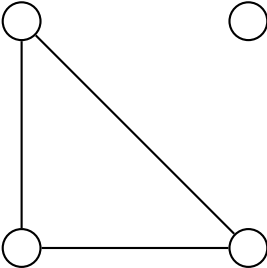


FIGURE 5. A disconnected graph.

Notice there are two parts to this graph. We call each of these a *connected component*. A connected component is a maximal connected induced subgraph; that is we cannot add any more vertices to the subgraph without disconnecting the graph).

Unless otherwise stated, we will assume all graphs discussed in this thesis are connected.

2.2. Basic Types of Graphs. A *complete graph* on n vertices, written K_n , is the graph such that there is an edge between every vertex, and all other vertices. The graph K_n has n vertices and $\frac{n(n-1)}{2}$ edges. Complete graphs are sometimes called *clique graphs*. This term is used more frequently when talking about subgraphs.

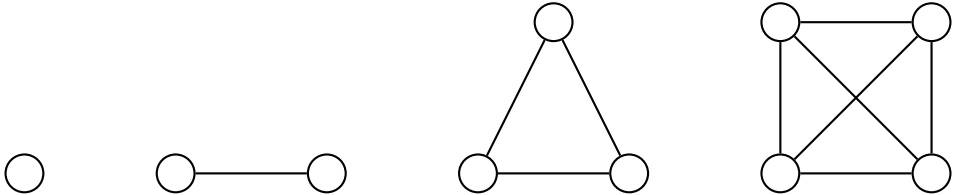


FIGURE 6. The graph K_n for $n = 1, 2, 3, 4$.

An *independent set* on n vertices is the graph G with $|V(G)| = n$ and $|E(G)| = 0$.



FIGURE 7. Independent set on 4 vertices.

A *path* on n vertices, written P_n , is defined as the graph with $V(G) = \{v_1, \dots, v_n\}$ and with edges $\{v_i, v_{i+1}\}$ for $0 \leq i \leq n - 1$. The graph P_n has n vertices and $n - 1$ edges.

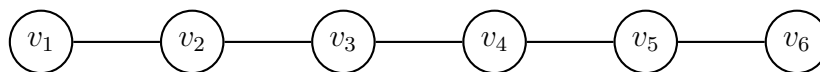


FIGURE 8. The graph P_6 .

A *cycle* on n vertices, written C_n , is defined as the graph with $V(G) = \{v_1, \dots, v_n\}$ and with edges $\{v_i, v_{i+1}\}$ for $1 \leq i \leq n - 1$ and the edge $\{v_n, v_1\}$. It is just a circle of vertices and edges. The graph C_n has n vertices and n edges.

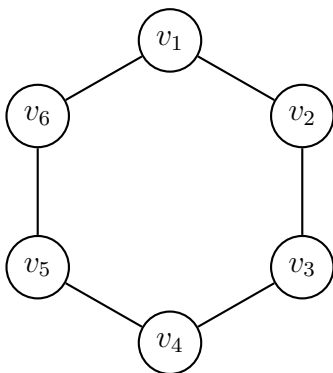


FIGURE 9. The graph C_6 .

We define a *bipartite graph* as a graph whose vertex set can be partitioned into non-empty sets X and Y such that there are no edges $\{u, v\}$

with u and v in the same partition. We define the *complete bipartite graph*, denoted $K_{m,n}$ with $|X| = n$, $|Y| = m$, as the bipartite graph that has all possible edges from X to Y . Note that a graph is bipartite if and only if it contains no odd cycle as a subgraph; see, for example, [22].



FIGURE 10. On the left a bipartite graph. On the right $K_{3,3}$.

2.3. Trees. A *tree* is a graph that is connected and has no cycles. A *forest* is a graph that has no cycles, but is not necessarily connected. That is, each connected component in a forest is a tree.

Each of the trees in the figure below is drawn in a manner so it appears “rooted.” This is a way of drawing a graph, called an *embedding* which is just an orientation of the graph on the plane. An embedding can also occur on a sphere, or a torus, or another topological surface, however that is beyond the scope of this thesis. In a rooted tree, we have a sense of vertices above, and other vertices below. We call the neighbour of a vertex v that occur above v , there is only one, the *parent* of v . All vertices below v , of which there are $(\deg(v) - 1)$ -many, are the *children* of v . This also gives us a notion of grandparents and cousins, although they are much less frequently discussed. Some vertices have only parents

and no children, thus they have degree one. We call these vertices *leaves*. We demonstrate this nomenclature in the following figure. Each root is labeled v , each leaf is labeled l , and we have labelled the parents and grandparents of the leaves p and g , respectively.

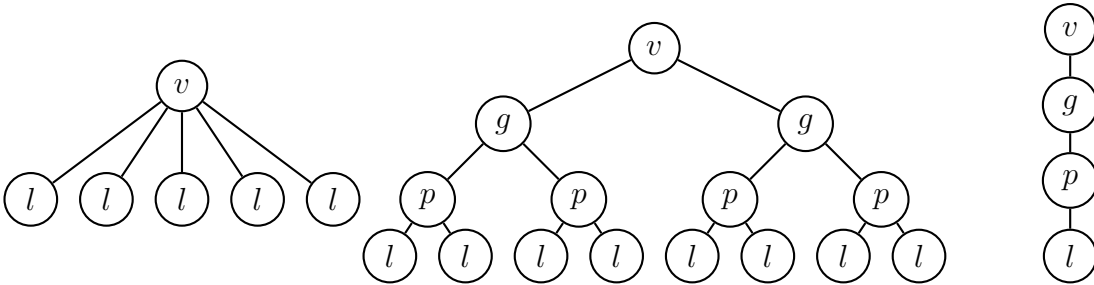


FIGURE 11. Examples of trees.

The tree in the middle is called a *binary tree*. Each vertex has at most two children. We can define an *m-ary tree* in a similar way by saying every vertex has at most m children. In the above example, this is a *complete binary tree* as every vertex, other than the leaves, has exactly two children.

To indicate at what point we have the leaves, we define the *height* of a tree. The root occurs at height 0, and we count down each level from there. That is, if an internal vertex, a vertex that is neither the root nor a leaf, x has height h , then the parent of x has height $h - 1$, and the children of x has height $h + 1$.

The tree on the left is the complete bipartite graph, $K_{1,5}$. We could draw it with the root in the middle, and all leaves on the outside so it looks like a *star*, which we generalize to be $K_{1,n}$. The tree on the right

is simply a path, we have just drawn it vertically. It is still a path. The difference between a path drawn horizontally, and a path drawn vertically, is just a different embedding in the plane. We say that a vertical path and a horizontal path are *isomorphic*.

Graphs $G = (V(G), E(G))$ and $H = (V(H), E(H))$ are said to be *isomorphic*, written $G \cong H$, if there is a bijection $f : V(G) \rightarrow V(H)$ such that any edge $\{u, v\}$ is in $E(G)$ if and only if $\{f(u), f(v)\}$ is in $E(H)$. Isomorphic graphs share all the same properties.

We include the following result on trees, whose proof can be found in any basic graph theory text; see [12, 22].

THEOREM 1. *Given a graph $G = (V, E)$ with $|V(G)| = n$, the following are equivalent.*

- 1) G is a tree.
- 2) G has $n - 1$ edges.
- 3) There is a unique path between any pair of vertices, $\{u, v\} \in V(G)$.

3. Graph Parameters and Domination

The *clique number* of a graph, denoted $\omega(G)$, is the maximum size of a subgraph H in G , where H is a clique. The *independence number* of a graph, denoted $\alpha(G)$, is the maximum size of a subgraph H in G , where H is an independent set, that is there are no edges in H . We define the *girth* of a graph to be the size of the smallest possible induced

cycle subgraph of G . We define a *dominating set* to be a set of vertices, $S \subseteq V(G)$ such that every vertex not in S is adjacent to a vertex in S . For every $v \in V(G) \setminus S$, there exists a $u \in S$ such that $\{u, v\} \in E(G)$. The *domination number* of a graph, denoted $\gamma(G)$, is the minimum size of a dominating set.

The following is the Petersen graph.

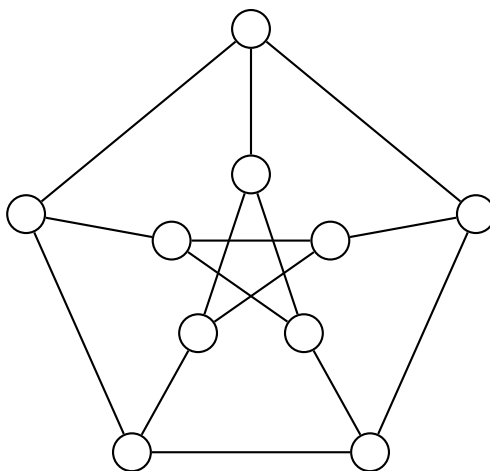


FIGURE 12. The Petersen graph.

In the Petersen graph G , $\alpha(G) = 4$. The girth of the graph is 5, and the largest clique is simply an edge, $\omega(G) = 2$. Further, $\gamma(G) = 3$.

In the game of Cops and Robbers, we focus on the parameter of *cop number*, denoted $c(G)$. Cop number is the minimum number of cops required to guarantee that the cops will capture the robber. To relate the previous parameters, we have that, for any graph G , $c(G) \leq \gamma(G)$. Cop number will be more thoroughly discussed in the following section.

The following bound on the domination number is part of folklore, and the proof is included for completeness.

THEOREM 2. For a graph G of order n , we have that $\gamma(G) \leq n - \Delta(G)$.

Proof: We begin by adding every vertex in $V(G)$ to a dominating set S . The vertex with degree $\Delta(G)$ we name as v . We can remove all neighbours of this vertex from S . Then the only vertices not in S are the ones adjacent to v . All of these vertices are adjacent to something in S , so S is still a dominating set. For any dominating set, S , we have $\gamma(G) \leq |S|$. Thus, $\gamma(G) \leq n - \Delta(G)$. \square

A *distance- k dominating set* is a set of vertices $S_k \subseteq V(G)$, such that every vertex not in S_k is connected to a vertex in S_k by a path of length no longer than k . We call the minimum size of all possible distance- k dominating sets to be the *distance- k domination number*, denoted $\gamma_k(G)$.

4. Introduction to Cops and Robbers

Cops and Robbers is a vertex pursuit game played on a graph, G . We often consider G to be a *reflexive* graph; that is, each vertex has a loop, this allows players to stay on their current vertex during their turn. The two players are the cop, C , and the robber R . In the 0^{th} round C places himself on some vertex of the graph, then R places herself on a vertex. Each player then takes one turn in each round beginning with the cop, by moving from their current vertex to an adjacent vertex. Since we play on a reflexive graph, staying on the same vertex is permitted.

In the original definition of the game in [17], only one cop is in play. However, we can also play with a set of cops C , where in each turn of the cops, all cops move simultaneously throughout the graph. The minimum number of cops required to capture the robber is called the *cop number*, denoted $c(G)$. Graphs with $c(G) = 1$ are called *cop-win*.

Cop-win graphs were first characterized for the finite case in [19], and later a characterization for all cop-win graphs was found in [17].

For any graph G , we define the relation \leq_r , for each $r \in \mathbb{N}$, on the set of vertices of G , $V(G)$, as follows:

- (1) $x \leq_0 y$ if and only if $x = y$
- (2) For each $r > 0$, we say $x \leq_r y$ if and only if $\forall v \in N(x), \exists u \in N(y)$ such that $u \leq_p v$ for some $p < r$.

We define \leq as \leq_α where α is the least non-negative integer such that $\leq_\alpha = \leq_{\alpha+1}$. This relation provides a characterization of cop-win graphs.

THEOREM 3 ([17]). *If for every $x, y \in V(G)$ $x \leq y$, then G is cop-win.*

More simply put, Theorem 3 says that no matter what starting position is chosen, the cop is able to choose a vertex so that wherever the robber chooses to move, the cop will be closer to the robber than he was previously. Eventually, the cop will capture the robber. This characterization is the basis for the main result in Chapter 4.

Since the above is a dense for an introduction to Cops and Robbers, we follow it with some examples (all known in folklore).

THEOREM 4. 1. For a graph G , we have that $c(G) \leq \gamma(G)$.

2. If G is a tree, then $c(G) = 1$.

3. For $n \geq 4$ and integer, $c(C_n) = 2$.

Proof. For item (1), note that every vertex in a dominating set is adjacent to every vertex outside the set. If the cops play on all vertices of a dominating set, then they will capture the robber on their first move.

For item (2), note that for a tree there exists a unique path between any pair of vertices $u, v \in V(G)$. The cop follows the unique path between himself and the robber. This path will never increase in length, and will eventually decrease until the robber is captured (via an inductive argument).

For the final item, observe that for a cycle, if a cop attempts to follow the robber, then they will simply go around the cycle in circles indefinitely. Thus, we need one cop to chase the robber in one direction, and a second cop to come around the cycle in other direction, eventually closing in on the robber from both sides. \square

5. A Brief Discussion on Algorithms

In Chapter 4 of this thesis we will discuss graph algorithms, and utilize Dijkstra's algorithm. An *algorithm* is a way of answering a YES or NO question given a certain input. Essentially, it is a set of steps that guides us without fail from an input to a certain output [9, 12, 22].

For example, you could create an algorithm to answer the question: “Is there a vertex of degree three in G ?” The algorithm could check the degree of each vertex and output YES if and only if there is a vertex of degree three. Such a vertex (if it exists) is a *certificate*. We denote this algorithm as follows.

DEGREE 3: Is there a vertex of degree three in G ?

The study of algorithms is a growing field of mathematics and computer science, and is far more complex than what is presented in this thesis. The basic knowledge discussed in this section is sufficient for the reader for the remainder of the thesis. For further discussion on algorithms and Cops and Robbers, the reader may consult [9].

We continue our discussion on algorithms with Dijkstra’s algorithm. We present the algorithm as follows.

DIJKSTRA: Is there a path between two vertices u and v ?

This algorithm gives a certificate of a shortest path whenever one exists. As input to the algorithm, we are given a graph G , a specified starting vertex u , an ending vertex v , and nonnegative edge weights.

- (1) Set $S = \emptyset$ to be the set of visited vertices, and the set $V(G) \setminus S$ to be the set of unvisited vertices. Assign tentative distances of ∞ to all unvisited vertices, and set the tentative distance of u to zero. Assign u to be the current vertex.

- (2) For each neighbour of the current vertex, take the distance of the current vertex and add the edge weight. If this value is less than the tentative distance of the neighbouring vertex, then reassign the tentative distance to this value. Otherwise, do nothing.
- (3) Once all neighbours of the current vertex have been visited, move the current vertex to S and remove the label of current vertex. Set some unvisited vertex to be the current vertex and repeat steps 2 and 3 until $S = V(G)$.
- (4) If the distance of v is finite, then there exists a path between u and v . Otherwise, u and v are in different components of the graph G .

6. Outline of Thesis

Now that we have presented an introduction to basic graph theory, as well as provided an overview of the original game of Cops and Robbers, we provide a summary of the thesis. In Chapter 2 where we will formally introduce Cops, Robbers, and Barricades. Here we will present basic results on the game with examples.

In Chapter 3, we will provide general bounds to the barricade-cop number of the graph. There will also be examples of modifications to the parameters of the original game. We introduce three modifications and show the large impact of small changes to the rules of the game.

In Chapter 4, a characterization of barricade-cop-win graphs will be given, along with an algorithm for determining the barricade-cop number of a graph. There will be a brief discussion on the complexity of these results. In the final chapter, we highlight our results and state open problems collected from the thesis.

CHAPTER 2

Cops, Robbers, and Barricades

1. An Introduction to the Game

It is a long tradition in the study of Cops and Robbers to examine variants of the game. There have been many variants that give power to the cops such as tandem-win Cops and Robbers, distance k Cops and Robbers, and photo radar Cops and Robbers [1, 5, 6, 11]. There have also been some variants that give power to the robber, such as Zombies and Survivors, and decoy Cops and Robbers. [8, 10, 14]. We define a new variant of the game of Cops and Robbers that gives more power to the robber. In this variant, the robber is able to change the structure of the graph in each round of the game.

DEFINITION 1.1. *Cops, and Robbers, and Barricades*, denoted CRB , is a game played on a reflexive graph G . We have two players: the set of cops, C , and the single robber R . Analogous to the original game of cops and robbers, in the initial round C places all the cops on some set on vertices of the graph, then R places herself on a vertex. Each player then takes one turn in each round beginning with the cops, each of whom, simultaneously, move from their current vertex to an adjacent vertex. Since we play on a reflexive graph, it is permissible for a player to remain

on the same vertex. Let b be a function $b : \mathbb{N}^+ \rightarrow \mathbb{N}$. The robber, on her turn, is allowed to move to any adjacent vertex, or instead of moving is permitted to build $b(k)$ many barricades on vertices, where k is the index of the current round. A *barricade* on a vertex forbids any player from entering that vertex. These barricades must be built in accordance with specified *barricade rules* which permit prescribed configurations of barricades to be built in each round.

In the present chapter, we will play with $b(k) = 1$ and we only consider the barricade rule that all barricades must be built on vertices adjacent to the robber. In Chapter 3, we will examine other barricade rules such as placing barricades on any vertex of the graph. In Chapter 4, we will consider the general case where $b(k)$ is an arbitrary function on the positive integers, and allow the building of barricades on any vertices.

The *barricade-cop number* of a graph, denoted $c_B(G)$, is the minimum number of cops required to guarantee capture. We say a graph is *barricade-cop-win* if $c_B(G) = 1$.

We first consider the following elementary but useful result.

LEMMA 5. For any graph G , $c(G) \leq c_B(G)$.

Proof. Given any graph G , and at most $(c(G) - 1)$ -many cops, we play CRB. The robber will chose to move each round and use the same strategy from the original game and never place a barricade in order to

evade capture indefinitely. Thus, we need at least $c(G)$ -many cops to capture the robber when playing CRB. \square

We can also find an upper bound for $c_B(G)$ by using a winning strategy for the cops via a dominating set. This result was first mentioned in Chapter 1, Theorem 4.1.

THEOREM 6. *For any graph G , $c_B(G) \leq \gamma(G)$, where $\gamma(G)$ is the domination number of G .*

Proof. The set of cops play on all the vertices of a dominating set of minimum size. For any initial placement of the robber, a cop will be able to capture R in the second round. \square

2. Preliminary Results

Now that we have a definition of the game, we consider $c_B(G)$ for some elementary classes of graphs. We first consider the path on eight vertices, P_8 . We notice $c(P_n) = 1$. We label the vertices on the graph such that v_1 is the left most vertex, and we count up as we move towards the other end of the path, so that the final vertex is labeled v_8 .

We begin by attempting the strategy used for paths in the original game using only one cop [9]. Then we show another example where the cops fail to win, which helps illustrate the subsequent proof.

In the first round, the cop will go on the middle vertex of the path, and then the robber will go to an end-vertex of the path to maintain as much distance between herself and the cop. In the next round, the cop will move one vertex closer to the robber. However, this time the robber will lay a barricade in the second-to-last vertex, protecting himself from the cop indefinitely. Thus, the robber wins.

Now, we attempt a strategy with two cops placed initially at each endpoint v_1 and v_8 . The robber will try to stay as far away as possible from both cops, and place herself close to the middle, on v_4 without loss of generality. In the first round, both cops move one step closer to the robber, to v_5 and v_7 , respectively. but the robber places a barricade to her left on v_3 , which prevents the cop on the left from ever reaching the robber. The robber will not choose to move since moving in either direction brings her closer to the cops. The cop on the right, is able to take another step toward the robber to v_6 . However, the robber is able to place another barricade on v_5 and has blocked both cops from ever reaching her, thus, winning the game.

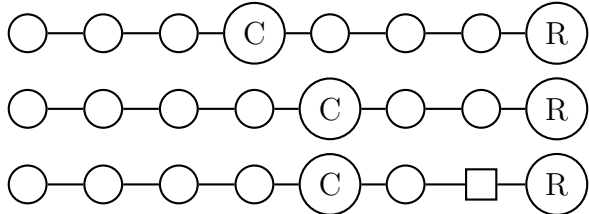


FIGURE 1. Example: Cops, Robbers and Barricades on P_8 .

In order for two cops to close in on the robber, we require that there must be less than six vertices in between two cops.

THEOREM 7. *For a path with $n \in \mathbb{N}^+$ vertices P_n , we have that*

$$c_B(P_n) = \left\lceil \frac{n+1}{6} \right\rceil.$$

Proof. We begin by providing a winning strategy for the cops, then we will count the required number of cops. We will show that this number of cops is best possible.

Label the vertices of P_n such that v_1 is the left most vertex, and counts upwards to the right. We place a cop on v_3 , and notice a cop here can guard v_1, v_2 . However, a cop starting at v_4 would not be able to guard v_1 , since R could build a barricade on v_2 in round 2, when the cop is still too far away at v_3 .

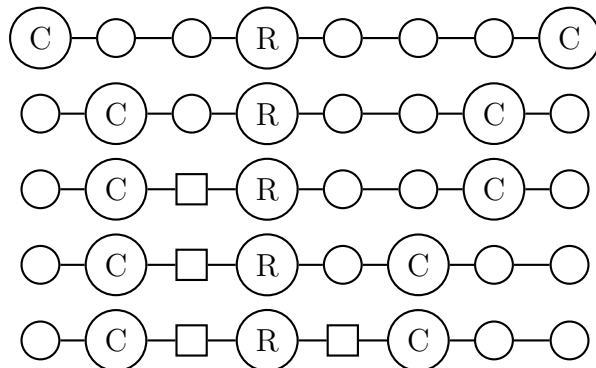


FIGURE 2. Example 2. The cops play on either end of the path.

From this point forward, we place a cop on ever sixth vertex. If R plays optimally, then she will place herself exactly halfway between two cops resulting in the following gameplay.

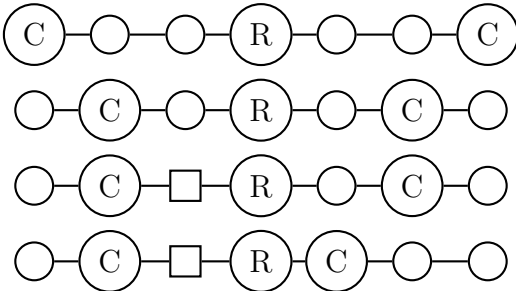


FIGURE 3. CRB on P_8 where the cops are placed optimally and win.

If we placed the cops further than every sixth vertex, then robber would have sufficient time to build both barricades before either cop is able to reach the robber. Thus, this is the minimum number of cops required.

When we reach the end of the path, we will have $n - 3 \pmod{6}$ vertices remaining to guard. If this is at most two, then our last cop will be able to guard the remaining two vertices for the same argument the first cop was able to guard v_1, v_2 . However, if there are three or more vertices remaining, then we will require an additional cop.

Therefore, we have a winning strategy for the cops by placing them six vertices apart (up to the arrangements on the leaves). Further, any other arrangement of cops will leave space for the robber to place a barricade on both sides and win the game. Hence, we have that

$$c_B(P_n) = \left\lceil \frac{(n-3)+4}{6} \right\rceil = \left\lceil \frac{n+1}{6} \right\rceil.$$

The proof follows. □

In the original game of Cops and Robbers, paths (which are a certain type of tree) are all cop-win. In the CRB variant, we need to guard a significant portion of the graph in order to keep the robber from winning. Note that c_B is distinct from the domination number on paths, which is $\lceil \frac{n}{3} \rceil$. The case for trees in fact becomes significantly more difficult and will be discussed in the following section. We next consider cycles.

THEOREM 8. *For a cycle with $n \in \mathbb{N}^+$ vertices C_n , we have that*

$$c_B(C_n) = \left\lceil \frac{n}{6} \right\rceil.$$

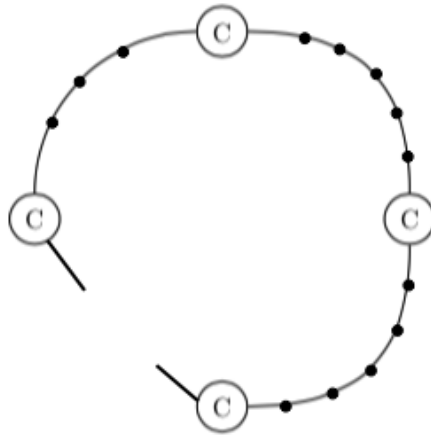


FIGURE 4. A cycle with a sample of the cop placement for CRB.

Proof. We follow the strategy used in Theorem 3. This time, there are no leaves, so we place a cop at some vertex of C_n , which we will call v_1 ,

then continue around the cycle placing a cop on every sixth vertex. At the end, we will always have at most five vertices between the last placed cop and the cop on v_1 . This results in a winning strategy for the cops. If we were to use fewer cops, then there would be a distance between some pair of cops that is greater than five, resulting in the robber winning by building herself into a win. \square

This new variant demonstrates that the barricade-cop number to be quite large, especially in comparison to the original cop number.

We define a robber-star as follows. A vertex $x \in V(G)$ is called a *robber-star* if for all vertices $v \in N(x)$, there exists a vertex $u \in N_{\deg(x)}(x)$ such that at least one path of length at most $\deg(x)$ from u to v passes through $N(x)$ uniquely through v . We call u a *star vertex*.

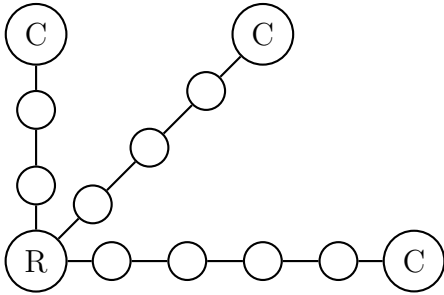


FIGURE 5. An example of a robber-star.

THEOREM 9. *If there exists a robber-star, then we require a cop on each star vertex.*

Proof. Suppose there is some star vertex u that does not have a cop. That is, there is some vertex adjacent to x that will remain unguarded after $\deg(x)$ many rounds. The robber R will play on x and build barricades on each neighbour other than the unique neighbour that corresponds to the path from the unguarded vertex u . Since there is no cop on u , on the $\deg(x)^{th}$ round, R will be able to build a barricade on vertex v , and win the game. \square

3. Trees

We begin this section with two theorems on general trees.

THEOREM 10. *For a tree T with set of leaves ℓ , we have that*

$$c_B(T) \geq |N_2(\ell)|.$$

Proof: All grandparents must be covered by cops in order to guard the leaves. If one leaf is not guarded, then R will place herself on that vertex. In the second round, the cops can move closer, then R can build and win. \square

THEOREM 11. *A tree T is barricade-cop-win if and only if there exists a vertex v such that, when T is rooted at v , T has height 2.*

Proof For a tree with height 2, we place a cop on the root of the tree. Since the robber is playing optimally she will place herself on a leaf in

order to lengthen the game play. In the first round, the cop will move down the unique path between himself and the robber. The robber will have no place to move or build a barricade that round and will need to stay put. The cop will capture the robber in the following round.

If a tree is barricade-cop-win, then we know that it can only have one grandparent of all the leaves. This grandparent is then a root for the tree such that the tree has height 2. □

4. Complete m -ary Trees

A *complete m -ary tree* with height h is a rooted tree such that each internal vertex has exactly $m+1$ children, and the leaves are at a distance of exactly k from the root.

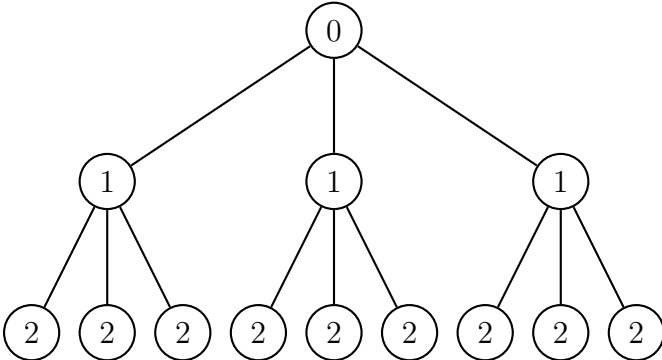


FIGURE 6. Complete 3-ary tree of height 2. The height of each vertex is labeled.

We consider the root to be at height 0, and the children of the root have height 1, we continue counting as such, then we will say leaves have height h .

THEOREM 12. *If T is a complete m -ary tree with height h , then we have that*

$$c_B(T) \leq \sum_{i=0}^c m^{(c-i)(2m+4)+m+1},$$

where $c = \lceil \frac{h-m-3}{2m+4} \rceil$.

Proof. We will place cops occupying all vertices of a given height, and they will all move together up or down the tree as a whole. We call the set of cops at height less than the robber the *above cops*, and we call the set of cops at a height greater than the robber the *below cops*. The strategy we are about to describe is called the *row strategy*.

To block the robber from entering a leaf on his first turn, we place cops at all vertices at height $h - 2$, of which there are m^{h-2} . If R were to enter a leaf in the first round, then the cops would move down in the next round and be adjacent to the robber. The only move the robber can make is to stay still as he has no where to build any barricades, thus, she will be captured by the cops in the second round.

If a robber enters any internal vertex on the first round, then there are m children and 1 parent for the robber to block off with barricades if he is going to win the game. Cops from the bottom must climb the tree to capture the robber before he has time to build all m barricades on the children; thus, if the robber is at height j the cops must be at height $j + m + 1$. Notice that it takes $m + 1$ robber moves to be totally

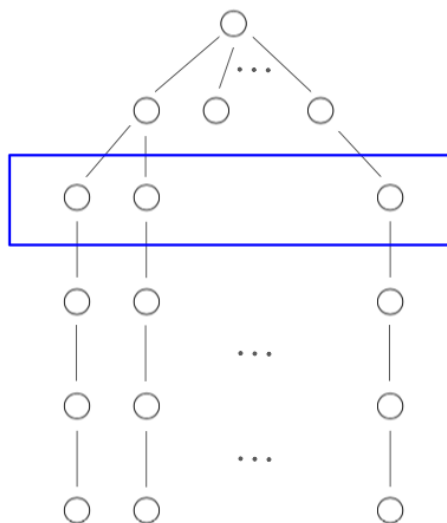


FIGURE 7. A row in a complete m -ary tree.

blocked in; thus, within the $(m + 1)^{st}$ move the cops must be adjacent to the robber so he can no longer build.

Consider the following two cases:

Case 1: If the robber builds a barricade above first, then the cops below are at a distance of m away. On the m^{th} turn, a cop will occupy the non-barricaded child of the robber, thus, R will be captured in the next round.

Case 2: If the robber builds a barricade below first, the cops above (specifically the cop that has R as a descendent) are now at a distance of m away. If R continues to only build below, the cop above will be adjacent to R in the m^{th} turn. However, if R at any point chooses to build above, he wastes a time-step, and the cops below are able to reach R before R can build all her barricades.

It is important to note that the robber will never chose to move. If she does chose to move, then she decreases the distance between herself and either the above cops or the below cops depending on her choice of movement. The respective set of cops is able to reach the cop in a small enough number of rounds so that R will not have sufficient time to build enough barricades to block herself off entirely before the cops are able to capture her. \square

This strategy will guarantee a win for the cops. However, it is unclear if the row strategy is the optimal strategy overall possible strategies.

THEOREM 13. *If the cops are forced to play the row strategy, then we have that*

$$c_B(T) = \sum_{i=0}^c m^{(c-i)(2m+4)+m+1},$$

where $c = \lceil \frac{h-m-3}{2m+4} \rceil$.

Proof. Here, it will not be possible for the cops to play any strategy other than the row strategy. In a sense we are altering the rules of the game. To show equality, we simply must show that removing a single cop from the game play presented above will result in R winning. There are two cases to consider in our analysis: when a cop is missing from *below* the robber, or when a cop is missing from *above* the robber.

Case 1: If a cop is missing from above the robber, then she will build all barricades below first. However, a sibling of the missing cop will move

up to the parent, and then back down to the missing location in order to try to capture the robber. This however, takes too much time ($m + 2$ moves) and R is able to build a barricade above himself first.

Case 2: If a cop is missing from below the robber, then we notice that in the first move of the cops, each vertex in that row has m cops on it, except the parent of the missing cop, which has only $m - 1$. There will still be at least one cop on the child of R after the m^{th} move, thus, the cops can win.

If there is only one cop missing from the row strategy, the robber is always able to find a vertex below the missing cop to place herself. Playing below the missing cop, guarantees a win for the robber. \square

Any tree of height h with maximum degree $m + 1$ is a subgraph of the complete m -ary tree of height h . Thus, an examination of these trees, will additionally provide an upper bound for any tree.

THEOREM 14. *If a tree T has height h and maximum degree $m + 1$, then we have that*

$$c_B(T) \leq \sum_{i=0}^c m^{(c-i)(2m+4)+m+1},$$

where $c = \lceil \frac{h-m-3}{2m+4} \rceil$.

Proof. We play the row strategy as above, however, we may be missing vertices in some rows. We add multiple cops to vertices in the same height, so that as they travel up or down the graph they act in the same manner as they did in the case of the complete m -ary trees. This will result in a win for analogous reasons. \square

We propose the following.

CONJECTURE 15. The row strategy is the optimal strategy for the cops when playing on m -ary trees.

We think the row strategy is optimal for m -ary trees, but we do not have a proof at the time of writing the thesis.

4.1. Covering Results for Trees. In the previous section, we discussed the row strategy for cops. Another possible strategy would be to have cops evenly placed throughout the graph. If the cops are evenly placed cannot win, then the cops could not win if they were not evenly placed. If they were not evenly placed, there would be a position of the robber such that the distance to any and all cops is maximized, thus, bigger than an even placement. With greater distance to the cops, the robber will have more time to build enough barricades and win.

We have $\Omega = \sum_{i=0}^{h-2} m^i$ many vertices that must be covered with Γ -many cops, where Γ is defined as follows,

$$\Gamma = \left(\sum_{i=0}^c m^{i(2m+4)+m+1} \right) - (m^{h-2}).$$

Hence, we must show $\frac{\Omega}{\Gamma} \geq m + 1$. This inequality guarantees the existence of a robber star, since each cop is further than the maximum degree $\Delta(G)$ from any other cop.

Suppose $\frac{\Omega}{\Gamma} < m + 1$. These values have been checked with a computer (see the Appendix for code and results) for $2 \leq m \leq 100$ and $10 \leq h \leq 1000$, most values cause no issues, and we do indeed have the proper conditions for the robber to win, however, there are a few exceptions (160 exceptions to be precise). Most of these exceptions occur when m and h get too close.

The minimum difference between m and h occurs at (7,10), and the maximum difference occurs at (99,303). These are the smallest and largest values in the list of exceptions. Further study is required to determine functions of m and h that make each bound optimal.

5. Locally Path-Like Graphs

DEFINITION 5.1. We call a graph G *locally path-like* if between every pair of vertices v_i, v_j with $\deg(v_i), \deg(v_j) \neq 2$ there are at least five

successive vertices with degree of two. We call this path of vertices of degree two a *local path*.

DEFINITION 5.2. We can represent locally path-like graphs with a weighted multi-graph, named $W(G)$. For each $v \in V(G)$ with $\deg(V) \neq 2$ we have a vertex, v' in $W(G)$. For any local path with endpoints u, v for length m , we add an edge u', v' in $W(G)$ with weight m . For a locally path-like graph G , we denote the edge set of $W(G)$ as $E(W)$.

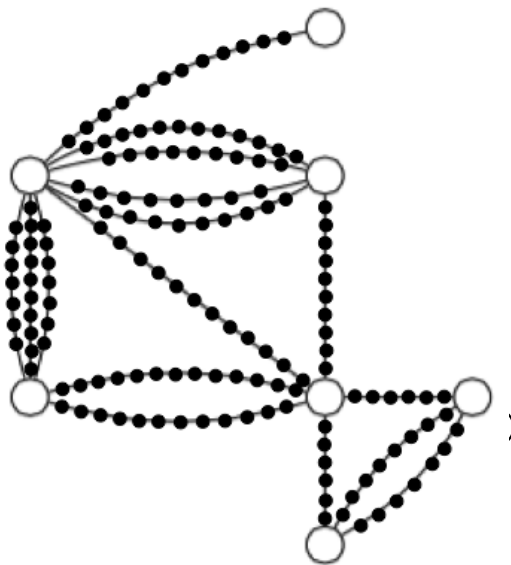


FIGURE 8. An example of a locally path like graph G , and its corresponding weighted multi-graph $W(G)$.

THEOREM 16. For a graph G we have that

$$c_B(G) \leq \sum_{e \in E(W)} \left\lceil \frac{w(e)}{6} - 1 \right\rceil + |V(W(G))|,$$

where $w(e)$ denotes the weight of an edge e in $W(G)$.

Proof. Initially, we place cops on all vertices in $W(G)$. For each edge $\{u, v\}$ in our new graph $W(G)$, they are paths in our original graph G . This path has a length of $w(\{u, v\})$ -many vertices in G . However, the first six vertices are guarded by the cop on u ; thus, we have a path of length $w(\{u, v\}) - 6$ to guard. We must guard every 6^{th} vertex. At the end of each local path, we will have at most five vertices left over, which will be guarded by the cop on v . The number of cops we require is $\left\lceil \frac{w(\{u, v\})}{6} - 1 \right\rceil$.

For each loop, $\{u, u\}$, we use the same strategy, and the final vertices are simply guarded by u as well. \square

6. Meyniel's Conjecture and Domination

6.1. Meyniel's Conjecture. One of the most famous conjectures in the study of Cops and Robbers is Meyniel's conjecture. In 1985, Meyniel conjectured in a personal communication with Peter Frankl, that the cop number for a graph G with at most n vertices must have that $c(G) = O(\sqrt{n})$ [9, 15]. Put more simply, there is some constant d , such that for any graph G , with at most n vertices, such that $c(G) \leq d\sqrt{n}$.

6.2. Domination Number. Recall from Chapter 1, the domination number of a graph is the minimum size of a set, $S \subseteq V(G)$, such that every vertex in G is either a member of the set S , or in the neighbour set of some vertex $u \in S$. We can denote this as $N[S] = V(G)$, where $N[S]$

is the union of the closed neighbour sets for each vertex $i \in S$. The set S is called a dominating set.

We can extend this definition to define the *distance k domination number*. First we define a *distance k dominating set*, $S \subseteq V(G)$ such that every vertex in G is either a member of the set S , or in the k^{th} neighbour set of some vertex $u \in S$. This is also to say, that for every $v \in V(G)$ there exists a $u \in S$ such that $v \in N_k[u]$. The minimum size of all distance k dominating sets is the *distance k domination number*, and it is denoted $\gamma_k(G)$.

In 2002, Sridharan et al. proved the following theorem.

THEOREM 17 ([21]). *If a graph G with $V(G) = n$ and no isolated vertices, then*

$$\gamma_2(G) \leq \frac{n - \Delta + 1}{2}.$$

6.3. Results. It has already been shown in Chapter 1 that for any graph $c(G) \leq \gamma(G)$ [9]. We prove the following new result.

THEOREM 18. *If G , with $V(G) = n$ has girth greater than or equal to γ , then*

$$c(G) \leq \gamma_2 \text{ and } c_B(G) \leq \gamma_2.$$

Proof Consider the following figure.

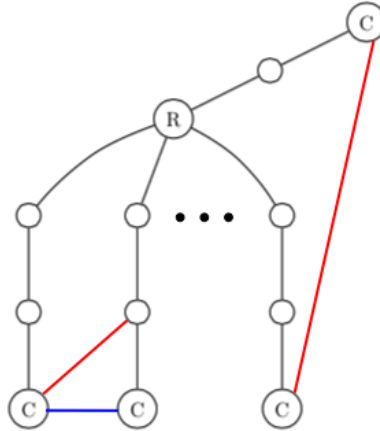


FIGURE 9. The initial set up. The blue edges may exist but are of no consequence to the proof, and the red edges cannot exist due to the girth of the graph being greater than or equal to 7.

In the initial set up, we place cops on each vertex of a distance 2 dominating set of minimum size. The robber is dominated by cops at most distance 2, and all of her neighbours are dominated by cops at most distance 2, as show in the above figure. For the first move of the cops, they will all move on the unique shortest paths connecting themselves and the robber (using that the girth is at least 7).

The robber must now move in order to avoid capture in the following round. If we are playing with barricades, then she must also move since she cannot lay a barricade without immediately being captured. However, if R moves to any neighbour she will be adjacent to a cop. Thus, R will be captured in the second round. Therefore, γ_2 -many cops can win when G has girth greater than or equal to 7. \square

THEOREM 19. *If G has $\Delta \geq (\sqrt{n} - 1)^2$, then $\gamma_2 \leq \sqrt{n}$.*

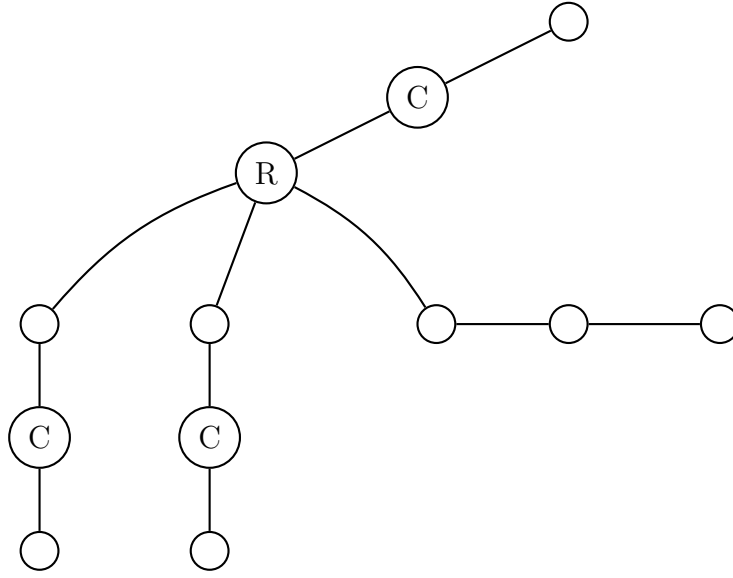


FIGURE 10. The first move of the cops, before the robber moves.

Proof. We note the following derivation, which follows from the bound in Theorem 13:

$$\begin{aligned}
 \gamma_2 &\leq \frac{(n - \Delta + 1)}{2} \\
 &\leq \frac{n - (\sqrt{n} - 1)^2 + 1}{2} \\
 &= \frac{n - (n - 2\sqrt{n} + 1) + 1}{2} \\
 &= \frac{2\sqrt{n}}{2} \\
 &= \sqrt{n}.
 \end{aligned}$$

The proof now follows. □

COROLLARY 20. *If G has girth greater than or equal to 7 and $\Delta \geq (\sqrt{n} - 1)^2$, then $c(G) \leq \sqrt{n}$.*

Proof. The proof follows by Theorems 14 and 15. □

CHAPTER 3

Variations of the Barricade Rules

1. Introduction

In the previous chapter we restricted ourselves to the following rules: the robber was allowed to build $b(k) = 1$ barricade each turn, and with the barricade rule of building only on adjacent vertices. Recall that the barricade rules dictate where the robber is permitted to build each turn. In this chapter, we will examine the changes in the game when we vary the barricade rules of the game, while keeping $b(k) = 1$. There are three particular examples that we will define and provide examples. We conclude the chapter by examining the relationship between the three variants.

2. Three Variants of the Barricade Rules

2.1. Global Version. We begin this section with the *global* version of the barricade rules. With the global barricade rules, the robber may place a barricade on an unoccupied and non-barricaded vertex anywhere on the graph. We denote the barricade-cop number of this version as c_B^1 . This variant makes it easier for the robber to barricade vertex cuts

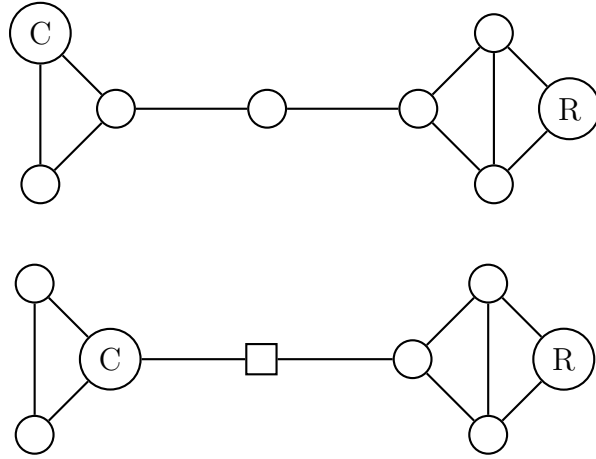


FIGURE 1. Example of the global variant.

without requiring her to move. Consider the following example with the given initial positions for both the cop and the robber.

First, the cop will move one step towards the robber. However, on her turn, the robber will build a barricade on the cut-vertex that separates herself from the cop. This results in a win for the robber.

We make the following definition required for further study of this variant. A *robber cut* is a set of vertices that separates the set of cops from the robber. Notice, that a robber cut is not necessarily a minimum vertex cut. Given a graph G , if there exists a robber cut of size k , where $k < \delta(G \setminus [N(C)])$, then the global barricade rules provide an advantage to the robber.

THEOREM 21. *Given a graph G with a particular starting placement of cops and the robber such that there exists a robber cut of size k . If the minimum distance between all cops and any vertex in the cut is strictly*

greater than k , then the robber wins when playing with global barricade rules.

Proof. The cops must eventually cross the robber cut in order to capture the robber. Thus, on their first move at least one cop will move towards the cut. In her first move, the robber will lay a barricade on one vertex of the cut. In this fashion, the cops continue moving one step closer to the cut, while the robber barricades the successive vertices of the cut. This proceeds until the second-to-last-move. Since the distance from any cop to the cut is more than k , on their k^{th} turn no cop will be on a vertex in the cut. On her final turn, the last vertex of the cut is unoccupied, and the robber builds herself in, winning the game. \square

Theorem 1 gives rise to the notion that we must restrict cops from being too far from a robber cut at all times. However, it does not yield an efficient lower bound. There are $n \cdot \binom{n}{k}$ many starting positions for specified k many cops. In order to determine the optimal k we would need to check each starting position with each number of k cops. The number of starting arrangements grows quickly from n^2 , when $k = 1$ to $4^n / \sqrt{\pi n}$ when $k = n/2$ [20]. An algorithmic argument would be impractical for determining a lower bound in this manner.

2.2. Local Version. The *local version* of the barricade rules is exactly the rules we used in Chapter 2, the robber may only place barricades on adjacent vertices. We denote the barricade-cop number, $c_B(G)$ as c_B^2

for the remainder of the chapter. For further discussion on this version, please reference Chapter 2 of this thesis.

2.3. Slug Version. As one may have noticed from earlier discussion, the aforementioned barricade rules do not necessarily require movement by the robber during game play. In the *slug version* the robber must move to a vertex, then move off it in order to leave a barricade. Just like a slug, the robber leaves a trail not of biological matter, but of barricades. It is not necessary for the robber to leave a barricade on a previous vertex. Let us proceed with an example.

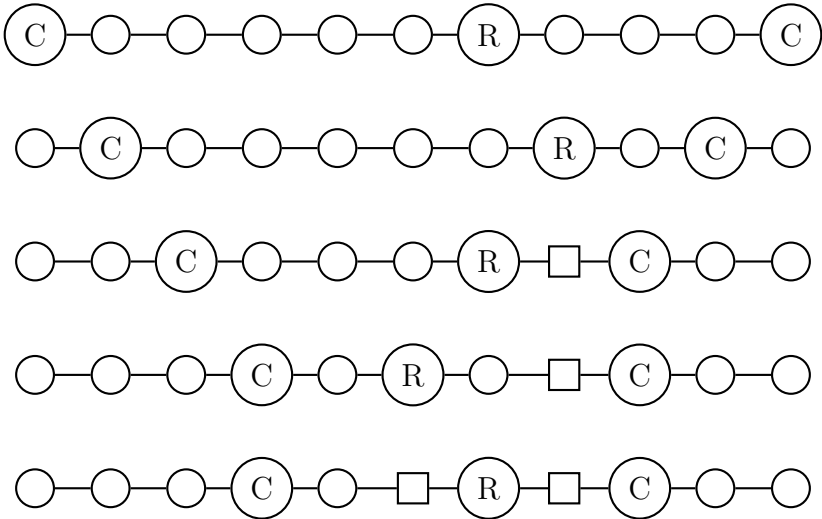


FIGURE 2. Example of the slug variant of CRB.

The cops and the robber begin in the given initial positions. Then as the cops move in, the robber moves first to her right and then to her left, placing barricades as she returns to her starting vertex. Notice that the robber builds barricades slowly, while the cops are able to quickly close in on the robber.

Consider the movement of the robber as she defends her starting vertex. In order to fully barricade herself in, it requires not $N(R)$ -many turns, but $2 \cdot N(R)$ -many. In the following example, we have each cop at a distance of at most $2 \cdot \delta(G) - 1$ from R . The robber then begins moving to a neighbour, building a barricade, and moving back to the starting vertex, v . On odd moves the robber will be on a neighbour and on even moves the robber will be on v . Round $2 \cdot \delta(G) - 2$ will end with the robber on v . Each cop has moved along its dominating path towards the neighbours of v . There will be exactly one neighbour that is free of barricades. The cop moves to that vertex trapping the robber on v , and capturing her in the next round.

Unfortunately, this does not give us an upper-bound for c_B^3 . In the second-to-last move as described above we would have the following set up.

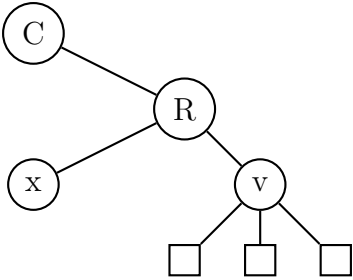


FIGURE 3. The second-to-last move in the slug version example.

Here, R is able to move to vertex x which is not necessarily currently guarded by cops.

3. Relating the Three Variants

Each of the previous variants of the game relate together in the following theorem.

THEOREM 22. *For any graph G , we have that*

$$c(G) \leq c_B^3(G) \leq c_B(G) = c_B^2(G) \leq c_B^1(G).$$

Below, we consider each of the inequalities in Theorem 2.

LEMMA 23. For any graph G , we have that

$$c(G) \leq c_B^3(G).$$

Proof. We require at least $c(G)$ -many cops when playing with the slug barricade rules. If we did not have $c(G)$ -many cops, then the robber would play her winning strategy in the original game. \square

LEMMA 24. For any graph G , we have that

$$c_B^3(G) \leq c_B^2(G).$$

Proof. Each time R builds in the local game, it takes R two moves to build the same barricade in the slug game (by moving and coming back to the initial vertex and thus, building the barricade). In the slug game, R can build the barricades in double the time of the local game, and this

gives the cops sufficient time to capture the robber. Thus, $c_B^2(G)$ -many cops can win in the slug version. \square

LEMMA 25. For any graph G , we have that

$$c_B^2(G) \leq c_B^1(G).$$

Proof. We show that $c_B^2(G)$ is a lower bound for $c_B^1(G)$ by contradiction. Suppose we have fewer than $c_B^2(G)$ -many cops in the global game. The robber can then use her winning strategy from the local game in order to win in the global game since it is possible to place barricades on only neighbouring vertices in the global game. \square

We now provide two examples, the first showing that we can have $c_B^2(G)$ equal to $c_B^1(G)$, and the second showing that it is possible for $c_B^1(G)$ to be strictly greater than $c_B^2(G)$. For this, we define a family of graphs called a *necklace graph*, denoted $N_i(G_1, \dots, G_i, k)$. We take any cop-win graphs G_1, \dots, G_i (not necessarily distinct) that are diameter k , and chain them together by overlapping them on a single vertex. More specifically, we choose a vertex u is in G_j and v is in G_{j+1} , then we delete v from the graph, and any edge that was incident to v we now make incident to u . For any set of graphs, we can choose a different set of vertices to chain them together, so for any set of graphs, we have a family of necklace graphs. We also make the restriction that each subgraph, or *pearl*, in the chain has minimum degree at least two. Below, we have an

example of a family of necklace graphs with diameter k , and six graphs in the chain.

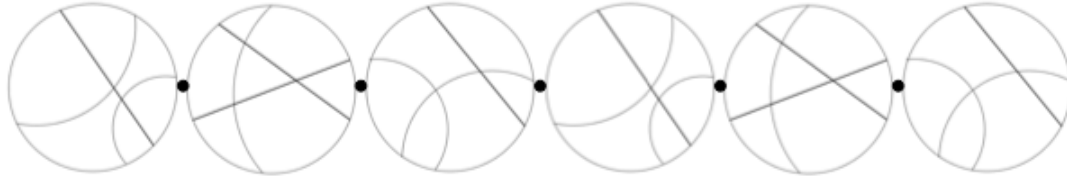


FIGURE 4. An example of $N_6(G_1, \dots, G_i, k)$.

In the following necklace graph, we have each graph in the chain is the same, with eight pearls. It is important to note that this graph is cop-win, and each pearl subgraph of the necklace is also cop-win.

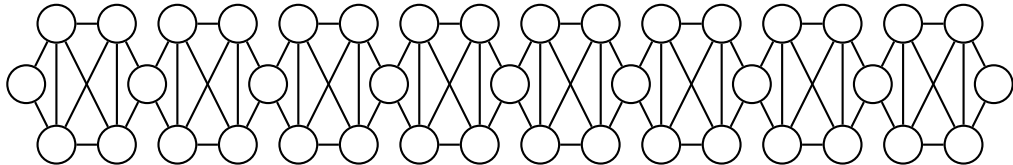


FIGURE 5. A particular necklace graph with eight pearls.

In order to show the necessary inequality, we will show that, in the graph G in Figure 3.5 $c_B^2(G) \leq 3$ and $c_B^1(G) \geq 4$.

To show that $c_B^2(G) \leq 3$, we first consider the starting positions of the cops. One cop must guard each end graph in the necklace by being on the chain vertex itself, or on a neighbour of the chain vertex in the adjacent graph. If not, then the robber would be able to build a barricade on the chain vertex in her first round. We place our third cop in the middle of the necklace, this minimizes the distance between this cop and the other two.

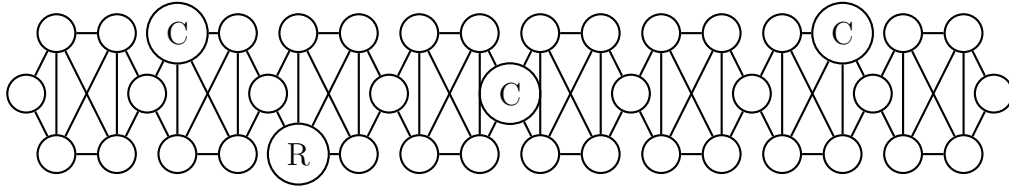


FIGURE 6. The initial positions for the local game of CRB on this graph.

We now consider the section of the graph beginning from the left-most cop to the right-most cop, for simplicity of the picture.

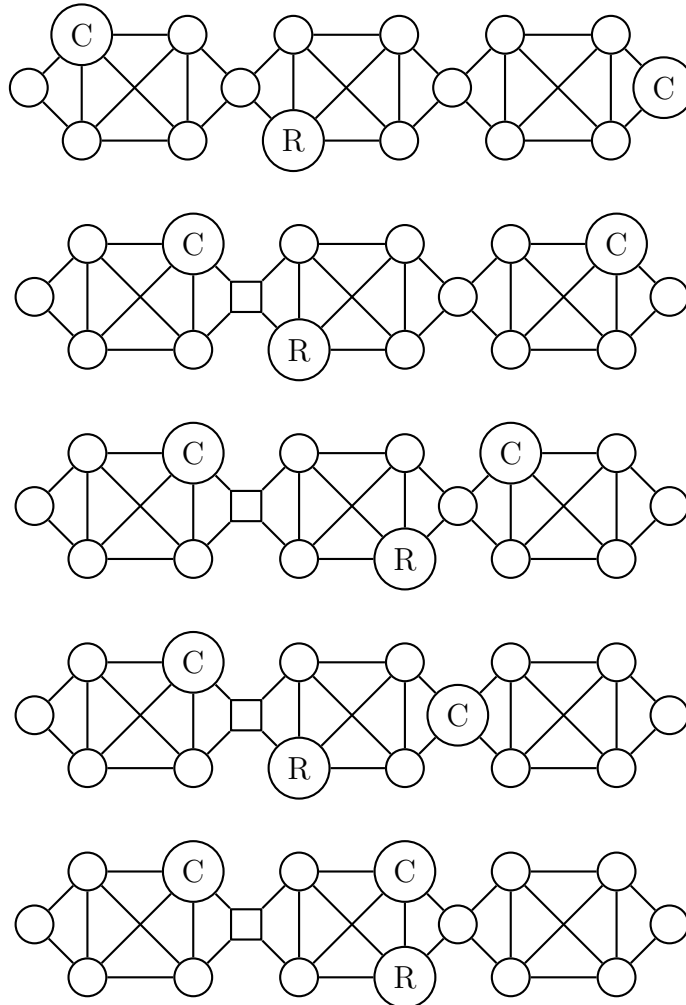


FIGURE 7. Game play of the local version of CRB on this particular section of a necklace graph.

In the second to last move of the cop, it is without loss of generality that the cop chooses to go to the top most vertex, since both vertices leave the robber stuck in a corner. Additionally, the second to last move of the robber is also symmetric. Following this last image, it is the cops turn, and the cop is able to capture the robber. Thus, we have that three cops are sufficient to win in the local version of Cops, Robbers, and Barricades on this particular graph.

For the next example, we must show that three cops is not sufficient to guarantee a win for the cops. We begin with the same initial set up, and initial positions for the cops and the robber, as in Figure 3.6. Notice that if there were a greater distance between any two cops, then the robber would still be able to win using the same strategy. If the robber can win when the cops play on the neighbour of the end chain vertices, then the robber can also win when the cops play on the chain vertices themselves. We will show the game play of the former, again on the section of the graph between two cops containing the robber.

At the end of the rounds in Figure 3.7, the robber has barricaded both chain vertices and there are no cops in the same component of the graph as the robber. Thus, R wins the game, and this demonstrates that $c_B^1(G) < 3$ but $c_B^2 = 3$ as required.

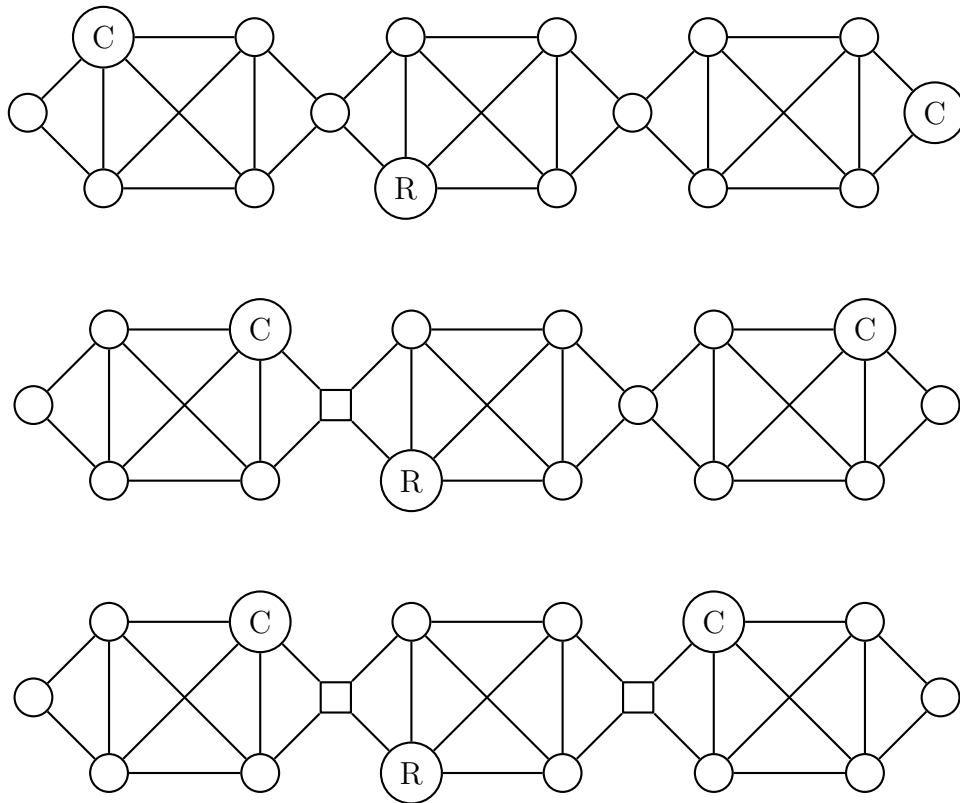


FIGURE 8. Game play of the global version of CRB, on this particular section of a necklace graph.

CHAPTER 4

Characterizing Barricade-Cop-Win Graphs

In the previous chapter, we examined the difference in game play when we varied the barricade rules, yet kept the barricade number constant at one barricade placed per turn. In this chapter, we will get examine a more general barricade number and provide a characterization for barricade-cop-win graphs. We begin with some useful notation.

DEFINITION 0.1. In the k^{th} round, there is some set of vertices that contain barricades; we call this set $B_k \subseteq V(G)$. That is, for each round k , we have a set B_k indicating which vertices of G contain barricades. Notice that $|B_k| \leq k \cdot b(k)$ since at most $b(k)$ -many barricades are being added each round. Additionally, we are not able to remove barricades from the graph, so for each k , we have $B_{k-1} \subseteq B_k$. Thus, $|B_K|$ is strictly increasing. We define the *barricade graph*, written G_k as the subgraph of G induced by $V(G) \setminus B_k$ for each round k ; that is,

$$G_k = G \langle V(G) \setminus B_k \rangle.$$

The barricade graph does not necessarily need to be connected. In the original game of Cops and Robbers, we restrict play to a connected graph. Here, we begin with a connected graph, but since B_k can be

any subset of $V(G)$, it is entirely possible that B_k contains a vertex cut, thus, leaving G_k disconnected. Of course, if C and R are in different components of G_k , then we could not have a barricade-cop-win graph. In previous examples we have seen what happens when there is a single cut vertex, such as when R plays on a leaf. In Chapter 3 we saw a scenario where there was a vertex cut of size two when we compared the local and global variants of the game. We now proceed with an example where there is a much larger vertex cut, and $b(k) = 2$. We start with the initial position of the cop being in the bottom left hand corner, and the robber being one vertex down the top left corner. The positions after the first round are shown.

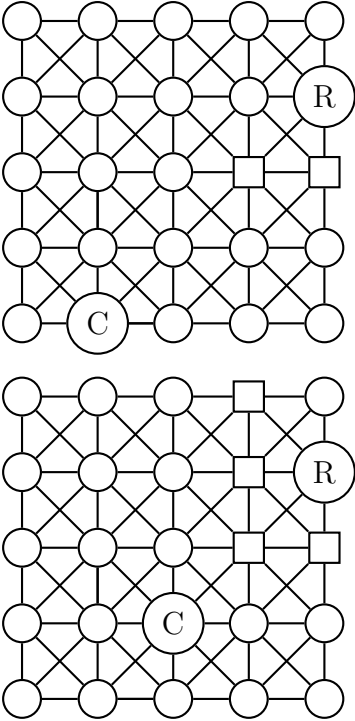


FIGURE 1. An example of CRB played on a 5×5 strong product grid with $b(k) = 2$ with a vertex cut of size four.

The robber in Figure 31 is now disconnected from the component of the cop. Consider the graph $G_k = G \setminus B_k$, shown in Figure 32.

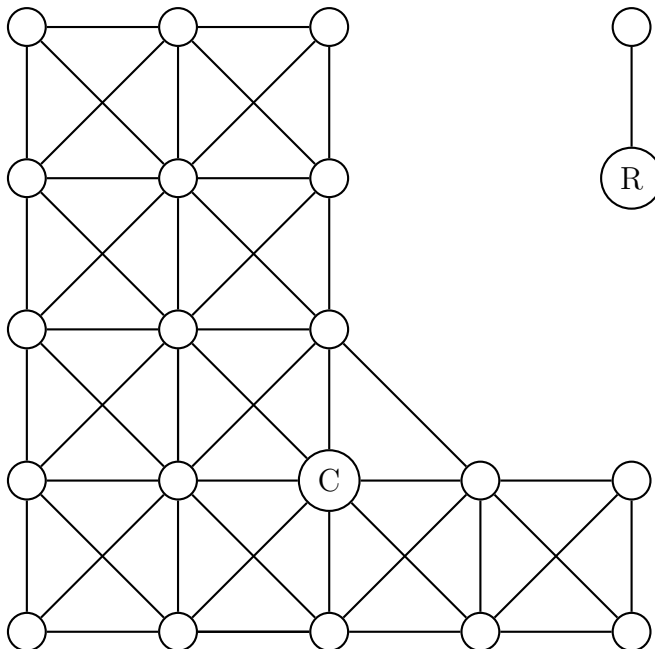


FIGURE 2. $G_k = G \setminus B_k$ for the example played in Figure 31.

Hence, we consider the minimum size of a vertex cut of G , and call it μ ; that is, G is μ -connected. If we are able to complete the game before $|B_k| \geq \mu$, then we will never have a disconnected graph during game play. We say that if some vertex v is in the same component as R in a graph G , then $v \in \text{comp}_G(R)$.

1. A relation on cop-win graphs

In this section we will define a relation that is analogous to the ones in [7, 17]. For any graph G_k we define the relation \leq_r , for each $r \in \mathbb{N}$, on $V(G_k)$ as follows:

- (1) $x \leq_0 y$ if and only if $x = y$.
- (2) For each $r > 0$, we say $x \leq_r y$ if and only if for all $v \in N(x)$, there exists a $u \in N(y)$ such that $u \leq_q v$ for some $q < r$.

Item 2 requires that for any position of the cop on a vertex y , and robber on a vertex x , for any movement of the robber from x to v , the cop is able to move to a vertex u in order to bring himself closer to the robber, where q is the number of moves left before capture.

Of course, we notice there will be some minimum non-negative integer α such that $\leq_\alpha = \leq_{\alpha+1}$. We let ρ be the largest of such smallest α .

2. Characterization

We derive the following relational characterization of barricade-cop-win graphs.

THEOREM 26. *A graph G is barricade-cop-win if and only if given any sequence of B_k , there is some vertex $u \in N_{G_{k-1}}(C)$ such that $u \in \text{comp}_{G_k}(R)$ for any k , such that $u \leq_\rho R$ for some finite ρ .*

Proof. We begin by proving the forward direction. Thus, we assume G is barricade-cop-win. From Lemma 2.1 we have that G must be cop-win in the original game, so in round $k = 0$, we have that $R \leq_\rho C$ for some ρ . Further, the cop is in the same connected component as the robber (recall our global assumption in Chapter 1 that we only consider connected graphs, unless otherwise stated).

Assume the result holds for a given fixed $k \geq 0$. In the k^{th} round we consider the move of the cop. Since we know he will eventually capture the robber, there must be some vertex $v \in N_{G_{k-1}}(C)$ that is in the same component of R that is not also in B_k . If the only such v were to also be in B_k , then all other vertices in the neighbour set of C are disconnected from R . Thus, R would remain free from capture indefinitely making G robber-win. After R makes her selection to move or add to B_{k-1} , we have C now on v and R on some vertex u both in the same component of G_k . Since C eventually captures R we must have some finite r , as defined in the relation, such that $u \leq_r v$ in G_k as required.

We now consider the other direction. Hence, assume that for any sequence of B_k , C is able to enter in G_{k-1} , the same component of R in G_k on some vertex u , for any k , such that $u \leq_\rho R$ for some finite ρ . We describe a strategy for C . There are two types of moves for C , which are based on whether or not R adds to B_k .

Type 1: In round k , if R adds to B_{k-1} , then by hypothesis, in round $k + 1$, C will move to the vertex u in $\text{comp}_{G_{k+1}}(R)$.

Type 2: In round k , if R does not add to B_{k-1} , but she instead moves to some vertex $v \in N_{G_{k-1}}(R)$ noting that $G_{k-1} = G_k$, then C is on some vertex $u \in \text{comp}_{G_{k-1}}(R)$ with $R \leq_r u$ for some finite r . Thus, for any choice of v there exists some vertex $y \in N_{G_{k-1}}(u)$ such that $v \leq_q y$ with $q < r$. In round $k + 1$, C moves to this y .

By making a combination of Type 1 and Type 2 moves, C will capture R by moving into $\text{comp}_{G_{k'}}(R)$, for some k' via R itself, or C will follow the strategy using the relations to capture R within its component. \square

3. Algorithm

As input for the algorithm we are given a graph G . The output of the algorithm will state whether the cop wins in the game of Cops, Robbers, and Barricades or not. In each round R is allowed to build $b(k) = \beta$ -many barricades, where β is a constant. Note that the first round of the game, round 0 has $B_0 = \emptyset$, thus, $G_0 = G$. There are between 1 and $(2n)!$ many choices of sequences (B_j) .

BARRICADE-COP-WIN: Is G barricade-cop-win?

Fix some sequence (B_j) . Assign a starting position to both C and R in round 0.

For each $i \geq 1$:

(1) If $B_{i-1} \neq B_i$, then R has added at most $b(k)$ -many new barricades:

(a) For each $u \in N_{G_{i-1}}(C)$ determine if there exists a path in G_i

from u to R as follows:

(i) In G , assign all edges to have weight one.

(ii) For every vertex $v \in B_i$, change the weight of all incident edges to ∞ .

(iii) Apply Dijkstra's Algorithm, discussed in Chapter 1, to find a shortest path between C and R , or determine no path exists.

Find u 's that are connected to R .

(b) Check if $R \leq_\rho u$ for some finite ρ , for some u from above.

(c) C moves to any such vertex, the choice is arbitrary.

(2) If $T_{i-1} = T_i$, then R did not delete a vertex and instead moved from v_R in G_{i-1} to $v'_R \in N_{G_{i-1}}(v_R)$ in G_i :

(a) We know there is some finite ρ such that $v_R \leq_\rho C$; thus, for any $x \in N_{G_i}(R)$ there is some $u' \in N_{G_i}(C)$ such that $x \leq_q u'$ for some $q < \rho$.

(b) Since R has moved to v'_R , there is some corresponding u' , to which C must move.

If for any sequence (B_j) we are always able to have $R \leq_0 C$ (that is, C captures R), then G is barricade-cop-win. If for any particular sequence $(B_{j'})$ we cannot capture the robber, then R will play that strategy and win. This algorithm verifies the necessary and sufficient conditions for barricade-cop-win graphs outlined in Theorem 4.1.

Note that the cop number of G_i for any i must be one. Otherwise, R would simply form G_i and then play her winning strategy in the original game.

4. Complexity

When discussing algorithms we would be amiss to omit a discussion on complexity. Although, it should be noted that a thorough analysis of the complexity of the preceding algorithm is outside the scope of this thesis.

We notice that there are an exponential number of choices for the barricade sequence. Hence, if given a sequence, then we are able to run the algorithm in polynomial time, but when having to create the sequences, that algorithm becomes exponential.

In the preceding algorithm there are most n steps, where $n = |V(G)|$. In each step, we apply Dijkstra's algorithm which is known to be in polynomial time [12, 22], or compute the relation which is also in polynomial time [7]. As we are given the barricades in advance, the complexity is bounded above by the sum of n polynomial functions, which in turn is polynomial.

However, when you are not given the sequence of barricades as input, it is necessary to compute the above for each possible barricade sequence, of which there is an exponential number. This yields an algorithm with exponential complexity. It may in fact be that determine if a given graph is barricade-cop-win without the barricade sequence as input is **NP-hard**. The determination of the complexity of the **BARRICADE-COP-WIN** problem is open.

CHAPTER 5

Conclusion and Open Problems

1. Conclusion

In this thesis, we introduced a new variant of the game Cops and Robbers called Cops Robbers and Barricades (or CRB). In Chapter 2, we introduced the game with a few examples, and provided the exact barricade-cop number for particular graph families. We presented an upper bound for m -ary trees that also extends as an upper bound to all trees. After defining locally-path-like graphs, we provided a discussion of CRB on those graphs. We concluded Chapter 2 by discussing conditions that ensure the cop number of a graph is at most \sqrt{n} .

In Chapter 3, we examined the impact of changing the barricade rules. We defined three new rules: the local rule where barricades can be placed only on neighbours, the global rule where barricades can be placed anywhere, and the slug rule where barricades can only be placed once the robber has moved away from a vertex. We considered the relationship between these games. In Chapter 4, we gave a characterization of barricade-cop-win graphs given any barricade rules and any $b(k)$ function.

2. Open Problems

We conclude this thesis by collecting a number of open problems on CRB.

- (1) We conjecture that the row strategy is the optimal strategy for the cops when playing on m -ary trees. The upper bound on c_B we proved for trees is likely not tight, since we currently use the bound for m -ary trees as an upper bound for all trees. Can we utilize graph packings to determine a lower bound for general trees?
- (2) In the version of the CRB game where we play with the slug variant, we would like to consider the possibility that the robber must build a barricade on every turn. This would imply the robber can never return to a previously visited vertex. How would this affect the barricade-cop number?
- (3) There is much to investigate in regards to the complexity of the algorithm presented in Chapter 4. What is the exact time complexity of computing c_B ? Would it be possible to determine if a graph is barricade-cop-win in polynomial time, or is the problem **NP**-hard?
- (4) Can we extend the barricade-cop-win characterization in Chapter 4 to the case of k -barricade-cop-win graphs, where $k > 1$?

- (5) A question that was not touched in the thesis would be to add randomness to this game. What is the barricade-cop number on the binomial random graph $G(n, p)$? When playing on deterministic graphs, what happens if the barricades are placed by the robber at random?
- (6) It would be interesting to play the CRB game on infinite graphs. For which graphs do we require an infinite number of cops? Is it possible for the robber to build enough barricades to reduce gameplay to a finite graph?

Appendix A - Code

3. m -ary tree bound

In this appendix, we present the code discussed in Chapter 2. We compare the two bounds of the m -ary trees, to determine which bound is smaller.

3.1. Code.

```
B = 1000 #upper bound for h

count = 0

max_d = 0
min_d = B

max_m = 0 # to keep track of where these differences happened
min_m = 0

max_h = 0
min_h = 0

print('Varying m and height:')
print('')

for j in range(10,B):
    h = j

    for i in range (2,100):
        m=i
        c = ceil((h-m-3)/(2*m+3))
        lhs = m^(c*(2*m+4) + 3*m + 6) + m^(3*m+5) - m^(c*(2*m+4) + 3*m \
+5) - m^(3*m+6)
        rhs = m^(h+2*m+4) + m^(h+2*m+3) + m + 1 - m^h - m^(h-1) - m^(2*m\
+5) - m^(2*m+4)
        if lhs<rhs:
            #print(True)
            print(m,h)
            count+=1
            d = h-m #difference between m and h
            if d > max_d:
                max_d = d
                max_m = m
                max_h = h
            if d < min_d:
                min_d = d
                min_m = m
                min_h = h
```



```
        break

print('end')
print 'count', count
print 'minimum difference', min_d, min_m, min_h
print 'maximum difference', max_d, max_m, max_h
```

3.1.1. *Results.*

Varying m and height:

```
(7, 10)
(2, 11)
(2, 12)
(10, 13)
(11, 14)
(3, 15)
(13, 16)
(14, 17)
(4, 18)
(2, 19)
(17, 20)
(5, 21)
(19, 22)
(20, 23)
(6, 24)
(22, 25)
(23, 26)
(7, 27)
(25, 28)
(26, 29)
(8, 30)
(28, 31)
(29, 32)
(9, 33)
(31, 34)
(32, 35)
(10, 36)
(34, 37)
(35, 38)
(11, 39)
(37, 40)
(38, 41)
(12, 42)
(40, 43)
(41, 44)
(13, 45)
(43, 46)
(44, 47)
(14, 48)
(46, 49)
```

(47, 50)
(15, 51)
(49, 52)
(50, 53)
(16, 54)
(52, 55)
(53, 56)
(17, 57)
(55, 58)
(56, 59)
(18, 60)
(58, 61)
(59, 62)
(19, 63)
(61, 64)
(62, 65)
(20, 66)
(64, 67)
(65, 68)
(21, 69)
(67, 70)
(68, 71)
(22, 72)
(70, 73)
(71, 74)
(23, 75)
(73, 76)
(74, 77)
(24, 78)
(76, 79)
(77, 80)
(25, 81)
(79, 82)
(80, 83)
(26, 84)
(82, 85)
(83, 86)
(27, 87)
(85, 88)
(86, 89)
(28, 90)
(88, 91)
(89, 92)
(29, 93)
(91, 94)
(92, 95)
(30, 96)
(94, 97)
(95, 98)
(31, 99)
(97, 100)

(98, 101)
(32, 102)
(33, 105)
(34, 108)
(35, 111)
(36, 114)
(37, 117)
(38, 120)
(39, 123)
(40, 126)
(41, 129)
(42, 132)
(43, 135)
(44, 138)
(45, 141)
(46, 144)
(47, 147)
(48, 150)
(49, 153)
(50, 156)
(51, 159)
(52, 162)
(53, 165)
(54, 168)
(55, 171)
(56, 174)
(57, 177)
(58, 180)
(59, 183)
(60, 186)
(61, 189)
(62, 192)
(63, 195)
(64, 198)
(65, 201)
(66, 204)
(67, 207)
(68, 210)
(69, 213)
(70, 216)
(71, 219)
(72, 222)
(73, 225)
(74, 228)
(75, 231)
(76, 234)
(77, 237)
(78, 240)
(79, 243)
(80, 246)
(81, 249)

(82, 252)
(83, 255)
(84, 258)
(85, 261)
(86, 264)
(87, 267)
(88, 270)
(89, 273)
(90, 276)
(91, 279)
(92, 282)
(93, 285)
(94, 288)
(95, 291)
(96, 294)
(97, 297)
(98, 300)
(99, 303)

end
count 160
minimum difference 3 7 10
maximum difference 204 99 303

Bibliography

- [1] M. Aigner, M. Fromme, A game of cops and robbers, *Discrete Applied Mathematics* **8** (1984) 1-11.
- [2] G.L. Alexanderson, About the cover: euler and konigsberg's bridges: a historical view. *American Mathematical Society Bulletin* **4** (2006) 567-573.
- [3] T. Andreae, Note on a pursuit evasion game played on graphs, *Discrete Mathematics* **9** (1984) 111-115.
- [4] A. Berarducci, B. Intrigilia, On the cop number of a graph, *Advances in Applied Mathematics* **14** (1993) 398-403.
- [5] A. Bonato, WHAT IS ... Cop Number? *Notices of the American Mathematical Society* **59** (2012) 1100-1101.
- [6] A. Bonato, E. Chiniforooshan, P. Prałat, Cops and robbers from a distance, *Theoretical Computer Science* **411** (2010) 3834-3844.
- [7] A. Bonato, G. McGillivray, Characterizations and algorithms for generalized cops and robber games, accepted to *Contributions to Discrete Mathematics*.
- [8] A. Bonato, D. Mitsche, X. Perez Gimenez, P. Prałat, A probabilistic version of the game of zombies and survivors on graphs, accepted to *Theoretical Computer Science*.
- [9] A. Bonato, R.J. Nowakowski, *The Game of Cops and Robbers on Graphs*, American Mathematical Society, 2011.
- [10] N. Clarke, Cops, Robbers, and Decoys, Preprint 2016.
- [11] N. Clarke, R.J. Nowakowski, Tandem-win graphs, *Discrete Mathematics* **299** (2005) 56-64.
- [12] R. Diestel, *Graph Theory*, Springer-Verlag, New York (1997).
- [13] L. Euler, Solutio problematis ad geometriam situs pertinentis, *Commentarii Academiae Scientiarum Imperialis Petropolitanae* **8** (1741) 128-140.
- [14] S.L. Fitzpatrick, J. Howell, M.E. Messinger, D.A. Pike, A deterministic version of the game of zombies and survivors on graphs, Preprint 2016.
- [15] P. Frank, Cops and robbers in graphs with large girth and cayley graphs, *Discrete Applied Mathematics*, **17** (1987) 301-305.

- [16] M. Maamoun, H. Menial, On a game of policeman and robber, *Discrete Applied Mathematics* **17** (1987) 307-309.
- [17] R.J. Nowakowski, P. Winkler, Vertex-to-vertex pursuit in a graph, *Discrete Mathematics* **43** (1983) 235-239.
- [18] D. Offner, K. Ojakian, Variations of cops and robber on the hypercube, *Australasian Journal of Combinatorics* **59** (2014) 229-250.
- [19] A. Quilliot, Jeux et pointes fixes sur les graphes, *Thèse de 3ème cycle*, Université de Paris VI (1978).
- [20] N.J.A. Sloane, Central binomial coefficients, *Online Encyclopedia of Integer Sequences* **A000984** (1991).
- [21] N. Sridharan, V.S.A. Subramanian, M.D. Elias, Bounds on the distance two-domination number of a graph, *Graphs and Combinatorics* **18** (2002) 667-675.
- [22] D.B. West, *Introduction to Graph Theory*, Prentice Hall, 1996.