The Cops and Robber game on some graph classes

by

Harmender Gahlawat

Advanced Computing and Microelectronics Unit

Indian Statistical Institute

Kolkata, India

A thesis submitted in partial fulfillment

of the requirement for the degree of

Doctor of Philosophy

in

Computer Science

Thesis Advisor: Prof. Sandip Das

January, 2022

Abstract

Cops and Robber is a two-player pursuit-evasion game played on graphs. Here, one player controls a set of k cops, and the other player controls a single robber. The game starts with k cops placing themselves on the vertices of a graph. More than one cops can occupy the same vertex. Then the robber enters a vertex of the graph. Then the game occurs in rounds and in each round, first the cops move, and then the robber moves. In a *cop move*, each cop either moves to an adjacent vertex or stays on the same vertex. In a *robber move*, the robber either moves to an adjacent vertex or stays on the same vertex. If at some point in the game, one of the cops occupies the same vertex as the robber, we call it a *capture*. The goal of the cops is to capture the robber, and the robber aims to evade the capture. The *cop number* of a graph G, denoted as $c(G)$, is the minimum number of cops that can ensure the capture against all strategies of the robber. A graph is *cop-win* if its cop number is 1. This game is a *perfect information game*, that is, each player can see other players and their moves.

Many variants of this game have been studied, mainly varying based on the capabilities of the cops and the robber. We begin our study by considering two variants: the *Cops and attacking Robber* and the *lazy Cops and Robber*, on various kinds of grids originating from the Cartesian product of paths and cycles. In particular, we study the *attacking cop number*

of planar grids, cylindrical grids, toroidal grids, high-dimenstional grids, and hypercubes. We study the *lazy cop number* of planar, cylindrical, and toroidal grids. We also study the classical Cops and Robber game on solid grids and partial grids.

Next, we study three models of this game on oriented graphs which differ based on the kind of moves the players can make. These models are (i) the *normal cop model*, where both cops and robber can only move along the direction of the arcs; (ii) the *strong cop model*, where the cops can move along or against the direction of the arcs while the robber can only move along them; and (iii) the *weak cop model*, where the robber can move along or against the direction of the arcs while the cops can only move along them. For the normal cop model, we show that there exist strongly connected oriented graphs having high girth, high minimum degree, and high cop number. We also characterize the cop-win graphs in various graph classes like transitive-triangle-free, outerplanar, and subcubic graphs. For the strong cop model, we construct graphs with unbounded cop number, and also study the cop number of grids, outerplanar, and planar graphs. For the weak cop model, we characterize the cop-win graphs.

Next, we study the game of Cops and Robber on string graphs. We give an algorithm to capture the robber on a string graph using at most 14 cops. Thus we prove that the cop number of string graphs is upper bounded by 14.

Let H be a subgraph of G. We say that cops *guard* H if the robber cannot enter the vertices of H without getting captured. Finally, we study the applications of guarding subgraphs to bound the cop number of some graph classes. In particular, we study the cop number of butterfly networks and AT-free graphs. We also study the game of *Cops and fast Robber* for AT-free graphs.

Contents

Listing of figures

To my parents.

Acknowledgments

When I started my PhD journey, I thought "I will do everything myself". But now, when I am writing my thesis, I do realise that there are more people I am grateful to than I can cover in this acknowledgements section. But I would still try to compile this surely incomplete list of people that helped me in this journey.

I would like to begin my acknowledgements by thanking my thesis supervisor, Professor Sandip Das. He is the main reason for my PhD experience being very pleasant and enriching. He gave me full freedom to choose the problems I wanted to work on and the people I want to work with. He always provided me with all the guidance I needed, both personal and professional. He is truly a friend, philosopher, and guide. I have learnt a lot from him during this time and hope to learn more from him in future.

Next, I would like to thank Dr Joydeep Mukherjee, Dr Soumen Nandi, and Dr Sagnik Sen. They provided me with guidance whenever I required it. I have learnt a lot from them. I thank Uma kant Sahoo for being my first helpline. He is the first person I tell my academic successes or failures. I also want to acknowledge Dibyayan Chakraborty, who shared with me invaluable tips on postdoctoral application process.

I am thankful to my labmate friends Subhadeep, Sanjana, Arun, Samim, and Saura for their warm friendship. I thank Juhi Chaudhary for her helpful suggestions whenever I needed them. I am also thankful to Ravindra, Subhadeep, and Mayur who helped me pave the road towards the beginning of my PhD.

I thank Dr Sandip Das for creating an environment which enabled me to meet Sagnik, Joydeep, Uma kant, Dibyayan, Shubhadeep, Ayan, Samin, Saura, Soumen, and Arun.

I am incredibly fortunate to learn from teachers like Dr Ansuman Banerjee, Dr Bhargab B. Bhattacharya, Dr Arijit Bishnu, Dr Sourav Chakraborty, Dr Nabanita Das, Dr Arijit Ghosh, Dr Sasthi C. Ghosh, Dr Krishnendu Mukhopadhyaya, Dr Subhas C. Nandy, Dr Sasanka Roy, and Dr Susmita Sur-Kolay.

Furthermore, I am grateful to my co-authors, from whom I have learnt a great deal of things. I am thankful to Sheikh Shakil Akhtar, Dr Dibyayan Chakraborty, Dr Sandip Das, Dr Florent Foucaud, Dr Subir Ghosh, Dimitri Lajou, Dr Bodhayan Roy, Uma kant Sahoo, and Dr Sagnik Sen.

I am also grateful to my thesis reviewers. They provided many helpful suggestions which improved the writing of this thesis significantly.

In drawing things to a close, I would like to thank my family for their untiring patience and infinite support. My parents, Nasib Singh Gahlawat and Suresh Gahlawat, always inspired and supported me to do whatever I chose to do. My wife Niyati Nehra, who was my girlfriend when I started my PhD, have been my continuous source of encouragement, inspiration, and perseverance. My brother Devender Gahlawat supported and motivated me during my PhD process. I am also very thankful to my parent-in-laws, Naveen Nehra and Babita Nehra, for their infinite patience and support. I feel much more gratitude towards my family than I can express here. I am also very thankful to my rest of family and friends that I could not mention here.

1

Introduction

Contents

The primary focus of this thesis is on the game of *Cops and Robber*. Before getting into the game, we give a brief introduction to the theory of games itself. Since the literature on the study of games is massive and beyond the scope of this thesis, we give a brief overview of the games that are relevant to the theme of this thesis.

In general, we consider a *game* as a structured form of play where a set of *players* contest, following some *rules*, to *win*. At any instance, the snapshot of the game is referred to as the *state* of the game. Some of these states are marked as *winning states*. Players change the state of the game aiming to reach a winning state. The player to reach a winning state first wins the game.

Although multiplayer games have been studied, we consider only the two-player games in this thesis. There are two major branches of study in the game theory: *combinatorial game theory* and *economic game theory*. Broadly, the combinatorial game theory deals with the games of *pure strategy* and *no chance*, and the economic game theory deals with games that also involve an element of chance. Humans started playing and discussing games long before the formal study of mathematical game theory began.

Combinatorial games are usually 2-player *sequential* games, where the two players move alternately. Moreover, these are *perfect information* games, that is, both players know the state of the game during the whole play. Also, there are no chance devices, and the game advances based purely on the strategy of the two players.

The study of combinatorial games began with the *impartial games*, in which both the players have the same allowable moves, and the moves allowed at any point in the game depend solely on the state of the game and not on which of the two players is playing. Some examples of impartial games are Nim [27], Sprouts [53], Kayles [45], and Cram [54]. In *partisan games*, the moves available to one player are not available to the other, that is, the moves allowed at any point in the game depend both on the state of the game and on which of the players is playing. One example is chess, where one player can move only the white tokens, whereas the other player can move only the black tokens. The games that we consider in this thesis are partisan games.

Although we do not consider the *economic game theory* in this thesis, it is worth mentioning that economic game theory encompasses an important part of game theory that deals with problems that arise in many practical scenarios. In economic game theory, there is an element of chance and there might be some chance devices involved. Moreover, in economic game theory, there might be *simultaneous play* where both players can act simultaneously (sequential games also exist). Furthermore, these might be *imperfect information* games, where players may not know the full state of the game all the time. The field of game theory is vast and the boundary between the combinatorial game theory and economic game theory is not very stringent. Also, it is worth mentioning that there are some variations of the Cops and Robber game that are studied in the framework of economic game theory. Some examples are *simultaneously moving Cops and Robbers* [74] and *generalized Cops and Robber* [70].

The games considered in this thesis are the pursuit-evasion games and fall broadly under the category of *graph searching*. Suppose we have to search for some data that is on some vertex of the graph. We can use any standard graph search algorithm like *breadth first search* (BFS) or *depth first search* (DFS), and search the data if it is on some vertex of the graph. However, if the *search object* is moving in the graph, these graph search algorithms do not ensure that they can search the search object. *Graph searching* encompasses *moving target search* in graphs, that is, the search object is moving in the graph. To capture the worst-case behaviour, we consider the search object as an adversary. Hence, we can define graph searching problems in the form of games between a set of searchers and some evaders that do not want to be found.

Suppose a person is lost in a cave and a group of searchers aim to find that lost person. Also, the searchers know the structure of the cave but do not know the position of the lost person. Also, the person who is lost has no coordination with the search team and may be moving in the cave. What is the minimum number of searchers that are required to search the lost person in the cave irrespective of the movement of the lost person? In the 1970s Richard Breisch, a spelunker, asked this question to his mathematician friend T.D. Parsons. Breisch also suggested some formulations of this problem and suggested some conjectures. Later, Parsons gave a precise mathematical formulation of this problem and wrote two papers [89, 90]. These papers gave rise to the formal theory of *graph searching* and *graph sweeping*. Parsons modeled this problem in the following manner. He considered the cave as a finite connected graph in which the lost person is moving continuously. The searchers must move according to a strategy based solely on the graph that must search the lost person, whose position is not known to the searchers. Moreover, to capture the worst-case behaviour, it was assumed that the lost person can see the movement of the searchers and does not want to be found. N. Petrov [91] studied a similar pursuit-evasion problem independently.

Golovach [59, 60] later proved that both these above problems, formulated by Petrov and by Parsons, are equivalent, and can be modeled as the following decontamination game. Let G be a graph. Initially, the whole graph G is *contaminated* and the goal is to *decontaminate* G. For this purpose, we have *searchers* and the goal is to use a minimum number of searchers. In a move, a searcher can either be placed on a vertex of the graph, removed from a vertex of the graph, or can *slide* from a vertex u occupied by the searcher to a vertex v along the edge uv. When a searcher slides along the edge uv, we say that the edge uv is *cleared*. If at some point in the game a cleared edge e can be connected with a contaminated edge using a path P that does not contain any searchers, then e becomes contaminated again. The goal is to clear all the edges of G using a minimum number of searchers. Here, the contamination may be considered as the *territory* of an arbitrarily fast invisible fugitive. Many variants of this game have been generated and studied by restricting or enhancing the abilities of the searchers or the fugitive.

In the above game model by Golovach, the fugitive can hide on both the vertices and the edges of the graph. These kinds of games naturally model problems like the following: searching for a moving vehicle on a road network, decontaminating a pipe system of some poisonous gas, and many more. The problems where the fugitive can hide on the vertices as well as the edges are usually referred to as the *graph sweeping* problems.

There are many practical problems which can be modeled by search games on graphs where the fugitive can only hide on vertices (although it can move along the edges). One such example is to look for some data on a computer network. The problems where the fugitive can only hide on the vertices of the graph are usually referred to as the *graph searching* problems. In all the games we consider in thesis, the fugitive can only stay on vertices, and hence all these problems fall under the category of graph searching.

Many variants of these problems are studied and these variants mainly differ on the following properties of the game.

1. Visible/Invisible fugitive: If the searchers can see the location and movement of the fugitive, then we say that the fugitive is *visible*, and *invisible* otherwise.

2. Speed of the fugitive: If the fugitive has speed s, then it can move along a path of length at most s that does not contain any searcher. Games with different speeds of the robber (ranging from $s = 1$ to unbounded speed) have been considered.

3. Lazy/Agile fugitive: The fugitive is *lazy* if it can move only if a searcher is attacking the fugitive, and is *agile* if the fugitive can move any time of the game.

4. Connected search: The search is said to be *connected* if the cleared region always induces a connected component.

5. Monotonicity of search: A search strategy is said to be *monotone* if the fugitive cannot recontaminate a cleared edge/vertex. It ensures that the searchers don't have to search an already searched subgraph. It is not always possible to find an optimal monotone search strategy. If the search strategy for a game is monotone, then we can check whether the game is in class NP by just considering the monotone strategies.

Many variants of the graph searching have been considered varying these properties and some other properties of the search. Alspach [4] wrote a brief survey on graph searching. Fomin and Thilikos [49] have compiled an annotated bibliography on graph searching, and Nisse [84] have compiled a recent survey on the topic.

Cops and robber is a graph searching game. In this game, the searchers are referred to as *cops* and the fugitive is referred to as the *robber*. This is a perfect information and sequential game. We define the game formally in Section 1.1.2. The cops as well as the robber can only stay on the vertices of the graph and make alternating moves starting with the cops. The goal of the cops is to capture the robber, and both cops and the robber have speed equal to 1. If at some point in the game one of the cops occupies the same vertex as the robber, we call it a *capture*. The goal of the game is to capture the robber using a minimum number of cops. Many variants of this game are also studied. A brief survey of the game is given in Section 1.2.

1.1 Preliminaries

In this section, we give the preliminaries required for this thesis. In Section 1.1.1, we define the standard graph theoretic terms and notions used in this thesis. In Section 1.1.2, we define the game of classical *Cops and Robber*.

1.1.1 Graph Theory Preliminaries

Graphs are primary object of study in this thesis. Hence, we begin by giving the necessary definitions and preliminaries about graph theory. For further details on graph theory, books by West [107] and Diestel [44] serve as good resources.

A *graph* G is a triple consisting of a *vertex set* $V(G)$, an *edge set* $E(G)$, and a relation that associates each edge with two (not necessarily distinct) vertices called its *endpoints*. A *loop* is an edge whose endpoints are the same, and *multiple edges* are the edges having the same endpoints.

A *simple graph* is a graph without loops or multiple edges. We denote a simple graph G as $G(V, E)$, having vertex set $V(G)$ and edge set $E(G)$. Here we consider $E(G)$ as a set of unordered pair of vertices and write an edge $e = uv$ (or $e = vu$) for an edge e with endpoints u and v. Unless specified otherwise, the graphs that we consider in this thesis are simple. Moreover, when it is clear from context, we denote $V(G)$ as V and $E(G)$ as E. The order of a graph is the size of its vertex set and is usually denoted by n, that is, $n = |V|$.

If u and v are two endpoints of an edge e, then we also say that u and v are *neighbours* and are *adjacent*. For a vertex u, we define its *open neighbourhood*, denoted by $N(u)$, as $\{v \mid uv \in E\}$. We define *close neighbourhood* of u, denoted by $N[u]$, as $N(u) \cup u$. We say that a vertex v is *dominating* or *universal* if $N[v] = V$. The *degree* of a vertex v, denoted by $d(v)$, is the number of edges incident on v. The *minimum degree* of a graph G is denoted by $\delta(G)$, and $\delta(G) = \min_{v \in V} d(v)$. Similarly, the *maximum degree* of a graph G is denoted by $\Delta(G)$, and $\Delta(G) = \max_{v \in V} d(v)$. A graph is said to be *k-regular* if each vertex has degree k.

A *homomorphism* f from G to H is a function $f : V(G) \longrightarrow V(H)$ (also denoted as $f : G \longrightarrow H$) which preserves edges, that is, if $uv \in E(G)$, then $f(u)f(v) \in E(H)$. An *isomorphism* f from G to H is a bijective function $f : V(G) \longrightarrow V(H)$ (also denoted as $f : G \longrightarrow H$) such that

$uv \in E(G)$ if and only if $f(u)f(v) \in E(H)$. We say that G is *isomorphic* to H, written $G \cong H$, if there is an isomorphism from G to H.

We say that a sequence of vertices $u_1, \ldots, u_k$ is a *path* if $u_i u_{i+1} \in E$ for $i < k$. Here we say that u_1 and u_k are the *endpoints* of the path. We say that P is a u, v-path if u and v are endpoints of P. Moreover, a path of order n is denoted as P_n. A graph is *connected* if there is a path between every pair of vertices in the graph.

A *cycle* on n vertices, denoted as C_n, is isomorphic to a graph with vertex set $V = \{v_1, v_2, v_3, \ldots, v_n\}$ and edge set $E = \{v_i v_{(i+1) \mod n)} \mid v_i \in V\}$. A graph is *acyclic* or *forest* if it contains no cycles. A *tree* is a connected and acyclic graph. The *girth* of a graph is the size of a smallest cycle of the graph.

A *chord* is an edge joining two non-consecutive vertices of a path or a cycle. A *chordless cycle* in G is a cycle of length at least 4 in G that has no chord. A *hole* is a chordless cycle in the graph.

We say that a graph $H(V', E')$ is a *subgraph* of graph $G(V, E)$, if $V' \subseteq V$ and $E' \subseteq E \cap (V' \times V')$. We say that H is an *induced subgraph* of G, if $V' \subseteq V$ and for $\forall u, v \in V'$, edge $uv \in E'$ if and only if $uv \in E$. Let x be a vertex of G. Then $G \setminus x$ represents the graph induced by vertices $V \setminus x$. Similarly, $G \setminus H$ represents the graph induced by vertices $V \setminus V'$. For a subgraph H of G, we define *close neighourhood of H*, denoted by $N[H]$, as $N[H] = \bigcup_{v \in V(H)} N[v]$.

For a graph G and $S \subseteq V(G)$, let $G[S]$ denote the *subgraph of G induced by the vertices of S*. A *complete graph* on n vertices, denoted by K_n, has n vertices which are pairwise adjacent. A subgraph H of G is said to be a *clique*, if H induces a complete graph. We say that a set of vertices S is *independent* if $G[S]$ contains no edges. A graph is said to be *bipartite*, if its vertices can be partitioned into two independent sets; and is said to be *k-partite* if its vertices can be partitioned into k independent sets. A *complete bipartite* graph, denoted by $K_{m,n}$, is a bipartite graph containing two partitions of size m and n such that two vertices are adjacent if and only if they are in different partitions.

Let $d(u, v)$ denote the length of a shortest u, v-path. If $u = v$, then $d(u, v) = 0$ and if there is no u, v-path, then $d(u, v) = \infty$. An *isometric* path is a shortest path between its endpoints.

Let $A \times B$ denote the Cartesian product of two sets A and B. The *Cartesian product* of two graphs G and H, denoted by $G \square H$, has vertex set $V(G) \times V(H)$ and vertices (a, a') and (b, b') are adjacent if $a = b$ and $a'b' \in E(H)$, or $a' = b'$ and $ab \in E(G)$. Several other graph products have been considered in the literature and the book by Imrich and Klavžar [66] serves as a good reference for the subject.

For a graph G, a set of vertices S is said to be a *dominating set* if every vertex not in S has a neighbour in S. A dominating set S with the minimum cardinality is said to be the *minimum dominating set* and the cardinality of S, that is $|S|$, is the *domination number* of G.

In this thesis, we consider directed graphs too, also referred to as *digraphs*. A directed graph is defined analogous to an undirected graph other than the fact that each edge is an *ordered* pair. The edges in directed graphs are referred to as *arcs*. Let uv be an arc in a directed graph $\overrightarrow{G}$, then u is referred to as the *tail* of the arc, v is referred to as *head* of the arc, and together both of them are referred to as *endpoints* of the arc. An arc where both its endpoints are the same is referred to as *loop*.

Let uv be an arc of a digraph $\overrightarrow{G}$. We say that u is an *in-neighbor* of v and v is an *out-neighbor* of u. Let $N^-(u)$ and $N^+(u)$ denote the set of in-neighbors and out-neighbors of u, respectively. A vertex without any in-neighbor is a *source* and a vertex without any out-neighbor is a *sink*.

An *orientation of a graph* G is an assignment of direction to the edges of the undirected graph G, resulting in the directed graph $\overrightarrow{G}$. Here G is said to be the underlying graph of $\overrightarrow{G}$. A *directed path* is an orientation of a path such that all arcs are oriented in the same direction. Similarly, a directed cycle is an orientation of a cycle such that all arcs are oriented in the same direction. A digraph is said to be a *directed acyclic graph* (also referred as DAG), if it does not contain a directed cycle. A digraph is said to be *strongly connected* if there exists a directed path between all ordered pairs of vertices.

An oriented graph is a digraph without loops or directed 2-cycles. Furthermore, assigning orientations to any simple graph gives an oriented graph. In this thesis, the directed graphs we consider are mostly oriented graphs.

1.1.2 Cops and Robber Game

Several variations of the Cops and Robber game have been studied. Here we define the variation given by Aigner and Fromme [3], also referred to as the *classical Cops and Robber*.

Cops and Robber is a two-player pursuit-evasion game played on a connected graph. Here, one player controls a set of k cops, and the other player controls a single robber. The game starts with k cops placing themselves on the vertices of a graph. More than one cop can occupy the same vertex. Then the robber enters the graph on a vertex (not already occupied by a cop). Then the game occurs in rounds and in each round, first the cops move, and then the robber moves. In a *cop move*, each cop either moves to an adjacent vertex or stays on the same vertex. In a *robber move*, the robber either moves to an adjacent vertex or stays on the same vertex. When on a cop/robber move, a cop/robber stays on the same vertex, we say that cop/robber *skips* the move or *passes* the move. If at some point in the game, one of the cops occupies the same vertex as the robber, we call it a *capture*. The goal of the cops is to capture the robber, and the goal of the robber is to evade the capture. If the cops can capture the robber in a finite number of moves, then we say that the cops win, else, the robber wins. Moreover, this is a perfect information game, that is, each player can see other players and their moves.

The *cop number* of a graph G, denoted by $c(G)$, is the minimum number of cops that can ensure the capture against all the strategies of the robber. If we place a cop on each vertex of the graph, then the robber cannot enter the graph without being captured. Hence the cop number is well defined for a graph G, and the order of G is the obvious upper bound for the cop number. Another obvious (but better) upper bound is the domination number of G, because if we place a cop on each vertex of a dominating set, then the robber will be captured in the next round (or the next cop move). Moreover, for a family $\mathcal{F}$ of graphs, $c(\mathcal{F}) = \max\{c(G) \mid G \in \mathcal{F}\}$. A graph is said to be *cop-win* if its cop number is 1. In this thesis, we represent the robber as $\mathcal{R}$.

The assumption of connected graph is obvious, as for a disconnected graph G having connected components $H_1, \ldots, H_m$, the cop number $c(G) = \sum_{i=1}^{i=m} c(H_i)$. Hence, the cop number of a disconnected graph G can be computed by computing the cop number of its connected components individually. Furthermore, the assumption of a single robber is also obvious, as if cops can ensure the capture of a single robber in a finite number of moves, then they can capture all robbers one by one in a finite number of moves. All graphs considered in this thesis are simple, connected, and finite. Although we consider the Cops and Robber game on only finite graphs, the game has also been studied on infinite graphs [20, 102]. Also, the cop number of an infinite graph can be finite [20].

The first question considered regarding this game was to characterize the cop-win graphs [86, 95]. Consider the last move of the robber $\mathcal{R}$ before the capture. Let $\mathcal{R}$ be on a vertex u and the cop $\mathcal{C}$ is at a vertex v. If $\mathcal{R}$ can move to a vertex x such that $x \notin N[v]$, then observe that $\mathcal{C}$ cannot capture $\mathcal{R}$. Hence, the capture is only possible if $N[u] \subseteq N[v]$. We call a vertex u a *corner* of v if $N[u] \subseteq N[v]$. Observe that for a graph G to be cop-win, G must have at least one corner vertex. A graph is said to be *dismantlable* if on successively removing a sequence of corner vertices, it can be reduced to a single vertex. It is known that a graph G is cop-win if and only if G is dismantlable [3, 86, 95]. Some examples of cop-win graphs are trees, K_n, and chordal graphs. Not all graphs are cop-win. For example, 2 cops are necessary to capture the robber in the graph C_4 (cycle on 4 vertices).

Let H and T be subgraphs of G. We say that cops *guard* H if the robber cannot enter the vertices of H without getting captured, by one of the cops guarding H, in the next cop move. We say that the robber is *restricted* to T, if the robber cannot leave the vertices of T without getting captured. Here T is the *robber territory*. Guarding a subgraph H of G is a technique used heavily to bound the cop number of graph G.

The *capture time* of a graph G using k cops, denoted as $capt_k(G)$, is the minimum number of rounds required by k cops to capture the robber in G, assuming the optimal play from both cops and robber.

1.2 Brief Survey

The game of Cops and Robber was introduced independently by Nowakowski and Winkler [86], and by Quillot [95]. All of them considered the game with a single cop and a single robber and characterized the graphs where a single cop can capture the robber. Later, Aigner and Frommer [3] generalized the game to multiple cops, which is now referred to as Classical Cops and Robber, and also defined the cop number. They showed that for a graph G having girth 5, $c(G) \geq \delta(G)$ and using this lower bounding technique showed that for every natural number k, there exists a graph G' such that $c(G') > k$. In contrast to this result, they proved that the cop number of planar graphs is at most 3, by proving and using the lemma that for a shortest u, v-path P of a graph G, one cop can guard P after a finite number of rounds. This lemma has been used heavily by a lot of authors and also in this thesis. Since the inception of this game, a lot of research has occurred in the field. The book by Bonato and Nowakowski [24] serves as a good reference on the topic.

A lot of research has been done regarding the computational complexity of finding the cop number of a graph. Berarducci and Inrigila [12] gave a backtracking algorithm that decides whether the cop number of a graph is at most k in time polynomial in n^k. Hence, for a fixed k (and a graph having cop number at most k), their algorithm implies a polynomial-time algorithm to compute the cop number of a graph. Goldstein and Reingold [58] considered the computational complexity of the Cops and Robber game (and some of its variants). They proved that, when k is a part of the input, it is EXPTIME-complete to decide whether k cops can win if either the initial positions are given or the graph is directed. They also conjectured that in the classical Cops and Robber game, to decide if k cops can win is EXPTIME-complete. Fomin et al. [46] proved the game to be NP-hard, that is, it is NP-hard to decide if k cops can capture the robber in a graph. Later, Fomin et al. [48] considered the PSPACE-hardness of some variations of the game under some restrictions. Mamino [81] proved that it is hard in PSPACE to decide whether k cops can capture the robber in a graph. Kinnersley [72] proved that it is EXPTIME-complete to decide whether k cops can win in an undirected graph, hence confirming a conjecture of Goldstein and Reingold [58]. Later, Kinnersley [73] showed that deciding whether k cops can win in a directed graph is polynomial-time equivalent to deciding whether k cops can win in an undirected graph, which gives an alternate proof of the EXPTIME-completeness of the Cops and Robber game.

Cop-win graphs were characterized in [3, 86, 95]. Later, Hahn and McGillivray [61] gave an algorithmic characterization of cop-win finite digraphs and then reduced the k cop game to 1 cop game, hence giving an algorithmic characterization of digraphs where k cops can win. Clarke and MacGillivray [38] gave a relational characterization of the graphs where k cops have a winning strategy, which in turn implies a slightly improved polynomial-time algorithm to decide whether, for a fixed k, k cops can win in a graph G.

Cop number is well studied in the context of the topological and geometric properties of a graph. Aigner and Fromme [3] proved that 1 cop can guard a shortest (isometric) path, and using this proved that the cop number for the class of planar graphs is 3. Later, Beveridge et al. [14] proved that 3 cops can prevent the robber to cross a shortest path in a unit disk graph, and applying techniques similar to Aigner and Fromme [3] proved that the cop number for a unit disk graph is at most 9. Gavenčiak et al. [56] used similar techniques and the idea of guarding the neighbourhood of a shortest path using five cops by Chiniforooshan [33], proved that the cop number of a string graph is at most 15. Gavenčiak et al. [56] also studied the cop number on various other intersection graph classes, such as outer-string graphs, interval-filament graphs, and string graphs of higher genus.

Aigner and Fromme [3] also proposed a question about the cop number of graphs embedded in the torus and orientable surfaces of higher genus and conjectured that one has to add 2 cops for going to higher genus. Quillot [96] proved the upper bound of this conjecture and showed that the cop number of a graph with genus g is at most $3 + 2g$. This result implies that the cop number of toroidal graphs is at most 5. Schroeder [97] later improved this result, refuting the lower bound of the conjecture, and showed that the cop number for a graph with genus g is at most $\lfloor 3/2 * genus(G) \rfloor + 3$. This also implies that the cop number for toroidal graphs is at most 4. Moreover, he proved that the cop number of a graph with genus 2 is at most 5. He also conjectured that for a graph G with genus g, $c(G) \leq 3 + g$. Bowler et al. [28] improved the upper bound further and proved that for graphs having girth g, the cop number is upper bounded by $\frac{4g}{3} + \frac{10}{3}$. Recently, Lehner [75] proved the tight bound and proved that the cop number for the class of toroidal graphs is 3, resolving a long-standing open question by Andreae [6]. Cop number of graphs that can be embedded in non-orientable surfaces has also been studied. Let G be a graph that can be embedded in a non-orientable surface having a non-orientable genus g'. Schroeder [97] gave an asymptotic bound proving that $c(G) = O(g')$. Later, Nowakowski and Schroeder [87] improved this upper bound and proved that $c(G) \leq 2g' + 1$. This bound was later improved by Clarke et al. [37], who proved that $c(G) \leq \frac{3g'}{2} + \frac{3}{2}$. Moreover, they proved that the cop number of a graph that can be embedded in a projective plane is at most 3, and of a graph that can be embedded in a Klein Bottle is at most 4. They also compared the cop number of graphs that can be embedded in orientable surfaces and that can be embedded in non-orientable surfaces. They showed that for a graph G having non-orientable genus g' and a graph H having genus $g' - 1$, $c(G) \leq c(H)$.

Another parameter studied for this game is the capture time. Bonato et al. [18] introduced the notion of capture time; and showed that the capture time for a cop-win graph, on n vertices, is at most $n - 3$ and for chordal graphs is at most $\lfloor \frac{n}{2} \rfloor$. Later, Gavenčiak [55] showed that for a cop-win graph of order n, the capture time is at most $n - 4$. Clarke et al. [36] gave an algorithm to compute the capture time of the cop-win graphs. Mehrabian [82] studied the capture time of two-dimensional Cartesian grids. Bonato et al. [19] showed that the capture time for a d-dimensional hypercube is $\theta(dlog(d))$. Pisantechakool and Tan [92] showed that the capture time for a planar graph with n vertices is at most $2n$. For a graph G on n vertices, having cop number k, it can be implied that capture time of G is $O(n^{k+1})$ (see [12, 18, 38]). Brandt et al. [30] and Kinnersley [73], independently, proved this bound to be tight for $k > 1$. They did so by constructing a family of graphs with n vertices, for all $k > 1$, such that the capture time of graphs from this family using k cops is $\Omega(n^{k+1})$.

Bonato et al. [25] studied the capture time of the graphs using more cops than required (more than the cop number of a graph). They showed that as more cops are added, the capture time decreases monotonically. They studied the capture times of trees, grids, hypercubes, and binomial random graphs using different number of cops. Breen et al. [31] introduced the notion of *throttling-number* of a graph, that is, the minimum value of $k + capt_k(G)$, where k is the number of cops used and $capt_k(G)$ is the capture time of G using k cops. They studied the throttling number of various graphs and characterized graphs having low throttling number. They also compared the behaviour of throttling number with various graph parameters such as domination number, radius, girth, and diameter of a graph. They also posed a question about the complexity of computing the throttling number, which was later answered by Shitov [100], who proved that computing the throttling-number of a graph is PSPACE-complete. Another interesting parameter to compute the speed-up using more cops was studied by Luccio and Pagli [79], where they defined *work* as $k \times capt_k(G)$ and the goal is to minimize the work. They studied the capture time for grids and tori using different number of cops.

The lower bounding technique of Aigner and Fromme [3] of girth five graphs is the most used technique for lower bounding the cop number of a graph. As implied by the lower bound for the *Zarankiewicz problem* [108], an extremal graph with girth at least five has $\Omega(n^{3/2})$ edges. In a graph with $\Omega(n^{3/2})$ edges if there is a vertex whose degree is less than $c\sqrt{n}$, for an appropriate constant c, then we can remove it and still get a smaller graph with $\Omega(n^{3/2})$ edges. Hence, eventually, every vertex has degree $\Omega(\sqrt{n})$. By the lower bounding technique of Aigner and Fromme [3], the cop number of such a graph is $\Omega(\sqrt{n})$. Meyniel [50] conjectured this to be tight, that is, $O(\sqrt{n})$ cops are sufficient to capture the robber in any connected graph. This is probably the deepest conjecture in this field, and has since been the focus of many results. Frankl [50] showed that for a graph G of order n, $c(G) = O(n\frac{loglogn}{logn})$. Later, Chiniforooshan [33] extended the idea of guarding a shortest path [3] to guarding the closed neighbourhood of a shortest path using five cops, and using this proved that $c(G) = O(\frac{n}{logn})$. Lo and Peng [78] proved the Meyniel's conjecture for graphs with diameter 2 and for bipartite graphs with diameter 3. Furthermore, for general graphs they showed that $c(G) = O\left(\frac{n}{2^{(1-o(1))\sqrt{log_2(n)}}}\right)$. Scott and Sudakov [98] also proved a similar bound for the Meyniel's conjecture independently, proving that $c(G) = O\left(\frac{n}{2^{(1+o(1))\sqrt{log_2(n)}}}\right)$. Frieze et al. [51] generalized these bounds to the variation of the game where the robber can move more than one edges at a time. The Meyniel's conjecture has also been studied for random graphs (see [16, 93, 94]). For a brief survey on Meyniel's conjecture, see the survey by Baird and Bonato [8].

Game of Cops and Robber is also studied on directed graphs. For directed graphs, the rules of the game are similar to the *classical Cops and Robber* game, with the only difference that a player (cop/robber), in its turn, can either move to an outneighbour of the vertex occupied by the player or stay on the same vertex. Hamidoune [62] considered the game on Cayley digraphs. Frieze et al. [51] studied the game on strongly connected digraphs and gave an upper bound of $O\left(\frac{n(\log\log n)^2}{\log n}\right)$ for cop number in digraphs. Along these lines, Loh and Oh [77] constructively proved the existence of a strongly connected planar digraph with cop number greater than three. They also proved that every n-vertex strongly connected planar digraph has cop number at most $O(\sqrt{n})$, thus confirming the Meyniel's conjecture on strongly connected planar digraphs. Hahn and MacGillivray [61] gave an algorithmic characterization of the cop-win finite digraphs. They also showed that any k-cop game can be reduced to 1-cop game, resulting in an algorithmic characterization for k-cop-win finite digraphs. However, these results do not give a structural characterization of k-cop-win or cop-win digraphs. Later Darlington et al. [40] tried to structurally characterize cop-win oriented graphs and gave a conjecture that was later refuted by Khatri et al. [71], who also studied the game in oriented outerplanar graphs and line digraphs. Kinnersley [73] showed that n-vertex strongly connected cop-win digraphs can have capture time $\Omega(n^2)$. Recently, the cop number of planar Eulerian digraphs and related families was studied in several articles [41, 63, 64]. In particular, Hosseini and Mohar [64] considered the orientations of integer grid that are vertex-transitive, and showed that at most four cops can capture the robber on arbitrary finite quotients of these directed grids.

De la Maza et al. [41] considered the *straight-ahead* orientations of 4-regular quadrangulations of the torus and the Klein bottle and proved that their cop number is bounded by a constant. They also showed that the cop number of every k-regular oriented toroidal grid is at most 13. Furthermore, Bonato and Mohar [23] explored some future directions of research in the directed graphs. Bradshaw et. al. [29] proved that the cop number of directed and undirected Cayley graphs on abelian groups has an upper bound of the form of $O(\sqrt{n})$. Modifying this construction, they obtained families of graphs and digraphs with cop number $\Theta(\sqrt{n})$. The family of digraphs obtained by this construction has the largest cop number in terms of n, of any known digraph construction.

Several variations of the game of Cops and Robber have been studied and they have various applications. The applications of the game extend from theory [99] to applications in artificial intelligence and other practical applications [4, 67, 105]. If the robber is invisible, the game is also said to be *zero-visibility Cops and Robber* and this game has been studied on various graph classes [42, 104]. The game when the robber is only visible if it is at a distance k is also studied and referred to as *limited visibility Cops and Robber* [35]. Another property is the *speed* of the robber and we say that the robber has speed s, if it can move along a path of length at most s. The model where the robber is faster than the cop is referred to as *Cops and fast Robber* [32, 43, 46].

One interesting variant is *Helicopter Cops and Robber* given by Seymour and Thomas [99], which characterizes the tree-width of a graph. In this model, the robber can move along any path in the graph that is not occupied by a cop. Also, each cop is either in a helicopter or on a vertex, and the goal of the cops is to land one of the cops from a helicopter on the vertex occupied by the robber. However, the robber can see the helicopter landing and can move from that vertex before the cop lands, evading capture. Seymour and Thomas [99] proved that a graph has cop number k for this model if and only if it has tree-width $k-1$.

They also showed that the search strategy is *monotone*, that is, the cops have a winning strategy in which the robber cannot occupy a vertex of the graph that was previously occupied by a cop.

Many variants of this game were later studied and were shown to have correspondence with various width parameters of the graphs. These variants vary on properties of the game like visibility, monotonicity, and whether the robber is agile or lazy. For example, if the search strategy is monotonic against an invisible and lazy (respectively agile) robber, then the cop number characterizes the tree-width (respectively path-width) of the graph [2].

Clean territory is the set of vertices that cannot be accessed by the robber at a given time step. We say that a search is *connected* if, at every step of the game, the clean territory induces a connected subgraph. More recently, researchers have focused on connected search. It is shown that, if the search is monotonic and connected against an invisible and lazy robber, then the cop number characterizes the connected tree-width of the graph. Also, if the search is monotonic and connected against an invisible and agile robber, then the cop number characterizes the connected path-width [2].

Researchers have generalized this game to directed graphs as well, to study the width parameters in directed graphs. In the directed graph model, the robber can move along a directed path not occupied by any cop. For a directed graph, if the search is monotone against a visible and agile robber, then the cop number characterizes the DAG-width of the graph [13]. If the search is monotone against an invisible and agile robber, then the cop number characterizes the directed path-width of the graph [11]. If the search is monotone against a visible and lazy robber, then the cop number characterizes the Kelly-width of the graph [65]. Apart from these, there are similar games that characterize other width parameters such as tree-depth [57], hypertree-width [1], cycle-rank[57], and directed tree-width [68].

1.3 Contributions and thesis overview

1.3.1 Models considered

In this thesis, we consider several variations of the Cops and Robber game other than the classical version defined above. They are as follows.

Cops and attacking Robber

The game of Cops and attacking Robber was introduced by Bonato et al. [17]. In this variant, the robber is able to *strike back* against the cops. If on a robber's turn, there is a cop in its neighborhood, then the robber attacks the cop and *eliminates* it from the game. However, if more than one cops occupy a vertex and the robber attacks them, then only one of the cops get eliminated, and the robber gets captured by the other cop. The cop number for capturing an attacking robber on a graph G is represented as $cc(G)$, and is referred to as *attacking cop number* of G. Bonato et al. [17] proved that $c(G) \leq cc(G) \leq 2 \cdot c(G)$. This can be verified easily as $cc(G)$ cops can capture the robber in the classical version; and if we play the attacking version with $2 \cdot c(G)$ cops using the strategy of the classical variant with the only difference that there are always at least 2 cops on a vertex. Aigner and Fromme [3] proved that for a graph G with girth at least 5, $c(G) \geq \delta(G)$ ($\delta(G)$ is the minimum degree in G). Bonato et al. [17] extended this result and proved that for a graph G with girth at least 5, $cc(G) \geq \delta(G) + 1$. They also proved that for an outerplanar graph G, $cc(G) \leq 3$. They further studied the relationship between $c(G)$ and $cc(G)$ and proved that for a bipartite graph G, $cc(G) \leq c(G)+2$. For a $K_{1,m}-$free diameter-2 graph G (with $m > 2$), they proved that $cc(G) \leq c(G) + 2 \cdot m - 2$.

Lazy Cops and Robber

The *lazy Cops and Robber* game was introduced by Offner and Ojakian [88], who also gave bounds for hypercubes. These bounds were later improved and generalized in [9, 101]. In this variant, at most one cop can move during the cop's turn. This restricts the ability of the cops with respect to the classical version. The cop number for lazy cops to capture the robber in a graph G is known as the *lazy cop number* and is denoted by $lc(G)$. Clearly, $c(G) \leq lc(G)$, as $lc(G)$ cops can capture the robber in the classical version (using the winning strategy of the lazy Cops and Robber game).

This variant is also referred as *one-cop-moves* game. Gao and Yang [52] studied this game on planar graphs and showed that there exists a planar graph with lazy cop number greater than 3. Wang and Yang [106] recently gave a relational characterization of graphs having cop number at most k. They also studied this game on graphs having treewidth 2, Halin graphs, and on the Cartesian product of trees having at least one edge.

Cops and Robber in oriented graphs

We consider three variants of the Cops and Robber game on an oriented graph $\overrightarrow{G}$. Since $\overrightarrow{G}$ is an oriented graph, one can define two types of moves. In a *normal move* the cop or the robber can move along the arc, whereas in a *strong move* the cop or the robber can also move against the arc.

In the *normal cop model*, Player 1 (controlling the cops) can perform at most one normal move on each of its cops, whereas Player 2 (controlling the robber) can perform at most one normal move on the robber. In the *strong cop model*, Player 1 can perform at most one strong move on each of its cops, whereas Player 2 can perform at most one normal move on the robber. In the *weak cop model*, Player 1 can perform at most one normal move on each of its cops, whereas Player 2 can perform at most one strong move on the robber.

The *normal (respectively, strong, weak) cop number* $c_n(\overrightarrow{G})$ (respectively, $c_s(\overrightarrow{G})$, $c_w(\overrightarrow{G})$) of an oriented graph $\overrightarrow{G}$ is the minimum number of cops needed by Player 1 to have a winning strategy in the normal (respectively, strong, weak) cop model. Furthermore, for a family $\mathcal{F}$ of oriented graphs

$$c_x(\mathcal{F}) = \max\{c_x(\overrightarrow{G}) \mid \overrightarrow{G} \in \mathcal{F}\}$$

where $x \in \{n, s, w\}$. Given a fixed model, an oriented graph is *cop-win* if Player 1 has a winning strategy playing with a single cop.

Cops and fast Robber

The game of Cops and fast Robber was introduced by Fomin et al. [46]. In this variant, the robber can move faster than the cops, that is, if the robber has *speed* s, then the robber can move along a path P of length at most s, such that no vertex of P is occupied by a cop. If $s = 1$, then this variant is equivalent to the classical Cops and Robber game. Here, *cop number* of a graph G, denoted by $c_s(G)$, is the minimum number of cops that are sufficient to capture the robber with speed s.

Cops and fast Robber game is well studied on graphs. The game was introduced by Fomin et al. [46]. They proved that this game is NP-hard and its parameterized version is W[2]-hard. They also showed that while for speed $s = 1$, the game is polynomial-time solvable for split graphs, the game becomes NP-hard for split graphs when $s = 2$. They also proved that the cop number for the class of planar graphs is unbounded when the robber is faster than the cops. Nisse and Suchan [85] studied the game of Cops and fast Robber on planar graphs and proved that $\Omega(\sqrt{logn})$ cops are necessary to capture a fast robber (even with $s = 2$) in $n \times n$ grid. Later, Balister et al. [10] considered this game on an $n \times n$ grid and improved this bound to show that $exp(\Omega(\frac{logn}{loglogn}))$ cops are necessary to capture a fast robber with high speed. Frieze et al. [51] studied this game and gave an upper bound on the cop number for a graph G. Dereniowski et al. [43] considered this game on interval graphs, with $s = \infty$, and showed that this game is polynomial-time decidable for interval graphs.

1.3.2 Organisation and Results

Chapter 2: Variations of Cops and Robber on Grids

In chapter 2, we consider the game of Cops and attacking Robber and the game of lazy Cops and Robber for various kinds of grids obtained from the Cartesian products of paths and cycles. Apart from that, we also study the game of classical Cops and Robber on the subgraphs of the planar grids. In particular, we have the following results.

In Section 2.3, we consider the game of attacking Cops and Robber on various kind of grids originating from the Cartesian product of paths and cycles. We show that the attacking cop number of planar grids is 2. Then we show that the attacking cop number of both cylindrical grids and toroidal grids is 3. Next, we consider the game of Cops and attacking Robber on hypercubes and show that in a d-dimensional hypercube $\lceil \frac{d}{2} \rceil +1$ cops are sufficient to capture the attacking robber. We also show that for a d-dimensional grid, d cops are sufficient to capture the attacking robber.

In Section 2.4, we present our results on lazy cops and robbers. We first show that the lazy cop number for both planar grids and cylindrical grids is 2. It seems that on toroidal grids ($C_m \square C_n$) a *few* lazy cops would not create problems for the robber. One of the ways to strengthen the ability of cops is to add a few *flexible* cops. In such a setting, all of the flexible cops are free to move in every round, while only one of the lazy cops can move in a round. We show that one flexible and two lazy cops can capture the robber on toroidal grids.

In Section 2.5, we consider the game of classical Cops and Robber on the subgraph of planar grids. We show that the cop number for the class of solid grids (defined in Section 2.5) is 2. Then we show that the cop number for the class of partial grids is 3.

Finally, we draw conclusions in Section 2.6.

Chapter 3: Cops and Robber on Oriented Graphs

In chapter 3, we consider the game of Cops and Robber in oriented graphs. The three graph parameters that we mainly consider are normal cop number $c_n(\cdot)$, strong cop number $c_s(\cdot)$, and weak cop number $c_w(\cdot)$ for oriented graphs. In particular, we have the following results.

We start our discussion by comparing the parameters $c_n(\cdot)$, $c_s(\cdot)$, $c_w(\cdot)$.

In Section 3.3, we study the normal cop number of oriented graphs. We begin by proving a Mycielski-type result by constructing oriented graphs with high normal cop number and girth. Then we attempt to characterize the cop-win oriented graphs in various graph families. It is easy to see that for an oriented graph to be cop-win, it must have a unique source. Therefore, all graphs that we consider for being cop-win are assumed to have a unique source vertex. In particular, we show that an oriented triangle-free graph is cop-win if and only if it is a directed acyclic graph (DAG). As a corollary, it proves that oriented bipartite graphs are cop-win if and only they are DAG. We also prove a similar result for outerplanar graphs, proving that an oriented outerplanar graph is cop-win if and only if it is a DAG. For subcubic graphs other than K_4, we show that an oriented subcubic graph (other than K_4) is cop-win if and only if they are DAG.

In Section 3.4, we study the strong cop model on oriented graphs. We begin by proving that there exist graphs with arbitrarily high strong cop number. We also extend this result to bipartite graphs. Next, we consider the strong cop number of oriented planar graphs, outerplanar graphs, and series-parallel graphs. In particular, we prove that the strong cop number of outerplanar graphs is two. We also prove that a specific class of oriented outerplanar graphs whose weak dual is a collection of paths are strong cop-win. We also consider the strong cop model on oriented Cartesian grids and show that they are also strong cop-win.

In Section 3.5, we consider the weak cop model on oriented graphs and characterize the weak cop-win oriented graphs. For this, we use a technique similar to the cop-win characterization by Nowakowski and Winkler [86].

Chapter 4: Cops and Robber on Intersection Graphs

In chapter 4, we consider the classical Cops and Robber game on some intersection graphs. In particular, we have the following results.

In Section 4.3.1, we consider the game of Cops and Robber on string graphs. We show that for a string graph G, 14 cops are always sufficient to capture the robber in G. This improves the result by Gavenčiak et al. [56] which gives a strategy to capture the robber in a string graph G using 15 cops. For that purpose, we also show that for a unique shortest u, v-path P, 4 cops can guard $N[P]$.

In Section 4.4, we consider the game of Cops and Robber on boxicity 2 graphs. We have the following result. Let *2-BOX* be the family of boxicity 2 graphs. Then $3 \leq c(2\text{-}BOX) \leq 14$. This improves the result by Gavenčiak et al. [56] which is $2 \leq c(2\text{-}BOX) \leq 15$.

Finally, we conclude our chapter in Section 4.5.

Chapter 5: Applications of Guarding subgraphs

For a subgraph H of graph G, we say that some cops guard H if $\mathcal{R}$ cannot enter H without being captured by one of the cops guarding H. This technique of guarding a subgraph has been used heavily in Cops and Robber games to find the cop number of various classes of graphs. We consider applications of guarding to find the cop number of some classes of graphs.

We begin our chapter by studying the Cops and Robber game on butterfly graphs and prove that the cop number for the class of butterfly graphs is two. We do so by using a novel guarding technique. Conventionally, in guarding a subgraph H of G, the subgraph H is connected and the cops guarding H stay on the vertices of H. For butterfly graphs, we guard a subgraph that induces an independent set using one cop and the cop enters the vertices of the guarded subgraph only to capture $\mathcal{R}$.

Next, we consider the game of Cops and fast Robber for graphs having a dominating pair (hence for AT-free graph too). Using an application of guarding the shortest path, we show that for a graph G having a dominating pair, $c_s(G) \leq s + 3$ improving upon the previously known bound, $c_s(G) \leq 5s - 1$, by Fomin et al. [46]. We also consider the classical Cops and Robber game on AT-free graphs and show that for an AT-free graph G, the cop number $c(G) \leq 3$.

Chapter 6: Conclusion

Finally, in Chapter 6, we discuss some open problems and future directions. Moreover, we introduce a new model for oriented graphs where some of the players have the ability to *push* the vertices of the graph. For a vertex v of an oriented graph, the *push* operation on v reverses the orientations of all the arcs incident on v. We define two kinds of push operations that can be performed by players: namely *weak push* and *strong push*. A player on vertex v having the ability to weak push can either move to an out-neighbour of v or can push v. A player on vertex v having the ability to strong push can either move to an out-neighbour of v or can push any vertex of the graph. Now, a player can have the ability to weak push, strong push, or no ability to push. Depending on what kind of abilities cops and the robber have, we can have 9 variations of the game, of which the one where neither the cops nor the robber can push is equivalent to the normal cop model.

2

Variations of Cops and Robbers on Grids

Contents

In this chapter, we study two variants of Cops and Robber games on several kinds of grids originating from the Cartesian product of paths and cycles. We also study the classical Cops and Robber game on subgraphs of planar grids. The two variants that we study are the Cops and attacking Robber and the lazy Cops and Robber. For both these variants, the cop number is lower bounded by the cop number of classical Cops and Robber game. In Section 2.1, we give a brief survey concerning this chapter and the definitions (and preliminaries) required for this chapter. In Section 2.2, we give an overview of the results included in this chapter. In Section 2.3 we consider the game of Cops and attacking Robber and in Section 2.4 we consider the game of lazy Cops and Robber. In Section 2.5 we consider the classical Cops and Robber game on solid grids and partial grids. Finally, we conclude in Section 2.6.

2.1 Preliminaries

2.1.1 Brief Survey

The game of Cops and Robber is well studied in the context of grids. Maamoun and Meyniel [80] considered the game of Cops and Robber on the product of trees and gave the cop number for the same. Their results imply that the cop number of a d-dimensional grid is $\lceil \frac{d+1}{2} \rceil$. Hence, the cop number for planar grids is 2. Neufeld and Nowakowski [83] extended this study to the products of various kinds of graphs. They also gave results for the cop number of products of trees and cycles. Their results imply that the cop number of cylindrical and toroidal grids is 2 and 3, respectively. Mehrabian [82] studied the capture time on planar grids. Luccio and Pagli [79] studied the capture time of planar, cylindrical, and toroidal grids and studied the capture time using more cops than the cop number of these grids. Several variations of the Cops and Robber game are also studied on grids. Nisse and Suchan [85] studied the game where the robber is twice as fast as the cops and showed that the cop number of planar grids is unbounded for this game. Balister et al. [10] studied the game of Cops and fast Robber on grids where the robber can be arbitrarily fast. Bhattacharya et al. [15] considered a variation of the game on d-dimensional grids where the robber has to move on each turn, and proved that the cop number here is d.

The game of Cops and attacking Robber was introduced by Bonato et al. [17]. In this variant, the robber can *strike back* against the cops. If on a robber's turn, there is a cop in its neighborhood, then the robber attacks the cop and *eliminates* it from the game. However, if more than one cops occupy a vertex and the robber attacks them, then only one of the cops gets eliminated, and the robber gets captured by the other cop. The cop number for capturing an attacking robber on a graph G is represented as $cc(G)$, and is referred to as *attacking cop number* of G. Bonato et al. [17] proved that $c(G) \leq cc(G) \leq 2 \cdot c(G)$. This can be verified easily as $cc(G)$ cops can capture the robber in the classical version; and if we play the attacking version with $2 \cdot c(G)$ cops using the strategy of the classical variant with the only difference that there are always at least 2 cops on a vertex. Aigner and Fromme [3] proved that for a graph G with girth at least 5, $c(G) \geq \delta(G)$ ($\delta(G)$ is the minimum degree in G). Bonato et al. [17] extended this result and proved that for a graph G with girth at least 5, $cc(G) \geq \delta(G) + 1$. They also proved that for an outerplanar graph G, $cc(G) \leq 3$. They further studied the relationship between $c(G)$ and $cc(G)$ and proved that for a bipartite graph G, $cc(G) \leq c(G) + 2$. For a $K_{1,m}-$free diameter-2 graph G (with $m > 2$), they proved that $cc(G) \leq c(G) + 2 \cdot m - 2$.

The game of lazy Cops and Robber was introduced by Offner and Ojakian [88], who also gave bounds for hypercubes. These bounds were later improved and generalized in [9, 101]. In this variant, at most one cop can move during the cop's turn. This restricts the ability of the cops with respect to the classical version. The cop number for lazy cops to capture the robber in a graph G is known as the *lazy cop number* and is denoted by $lc(G)$. Clearly, $c(G) \leq lc(G)$, as $lc(G)$ cops can capture the robber in the classical version (using the winning strategy of the lazy Cops and Robber game).

This variant is also referred as *one-cop-moves* game. Gao and Yang [52] studied this game on planar graphs and showed that there exists a planar graph with lazy cop number greater than 3. Wang and Yang [106] recently gave a relational characterization of graphs having cop number at most k. They also studied this game on graphs having treewidth 2, Halin graphs, and on the Cartesian product of trees having at least one edge.

2.1.2 Definitions

All graphs considered in this chapter are simple, connected, finite, and undirected. In this chapter, we refer to the robber as $\mathcal{R}$, single cop as $\mathcal{C}$, and multiple cops as $\mathcal{C}_1, \ldots, \mathcal{C}_k$. Let P_n and C_n denote the paths and cycles of order n respectively[1]. Let $\square$ denote the Cartesian product between two graphs. A planar grid (also referred as Cartesian grid), denoted as $P_m \square P_n$, is the Cartesian product of two paths P_m and P_n. A cylindrical grid, denoted as $C_m \square P_n$, is the Cartesian product of the cycle C_m and path P_n. A toroidal grid, denoted as $C_m \square C_n$, is the Cartesian product of cycles C_m and C_n. A d-dimensional grid, denoted as $P_{i_1} \square P_{i_2} \square \ldots \square P_{i_d}$, is the Cartesian product of the paths $P_{i_1}, \ldots, P_{i_d}$. The d-dimensional hypercube Q_d is defined recursively in terms of Cartesian products of two graphs as follows: $Q_1 = K_2$ and $Q_d = Q_{d-1} \square K_2$, where K_2 is an edge.

The vertex set of $P_m \square P_n$ is $\{(i,j) \mid 0 \leq i \leq m-1,\ 0 \leq j \leq n-1\}$ and vertex (i,j) is adjacent to $(i \pm 1, j), (i, j \pm 1)$ if they exist. $P_m \square P_n$ can also be visualized as a grid with n rows and m columns. A vertex of $P_m \square P_n$ is a *corner vertex* if its degree is two; a *boundary vertex* if its degree is three; and an *internal vertex* if its degree is four. The grid is oriented such that its four corner vertices $(0,0), (0,n-1), (m-1,n-1)$ and $(m-1,0)$ are arranged in clockwise order with $(0,0)$ at the bottom left. The boundary vertices are called *top, bottom, right* or *left* boundary vertices if the belong to the $n-1^{\text{th}}$ row, 0^{th} row, $m-1^{\text{th}}$ column or 0^{th} column respectively (see Fig. 2.1.1).

The cylindrical grid $C_m \square P_n$ is defined similar to $P_m \square P_n$, with extra edges between $(0,j)$ and $(m-1,j)$, for $0 \leq j \leq n-1$. The toroidal grid $C_m \square C_n$ is defined similar to $C_m \square P_n$, with extra edges between $(i,0)$ and $(i,n-1)$, for $0 \leq i \leq m-1$.

A graph is a *partial grid* if it is a subgraph of a planar grid. A graph is a *solid grid* if it has an embedding such that it is a subgraph of a planar

[1] For sake of convenience, we do not consider P_1 as a path.

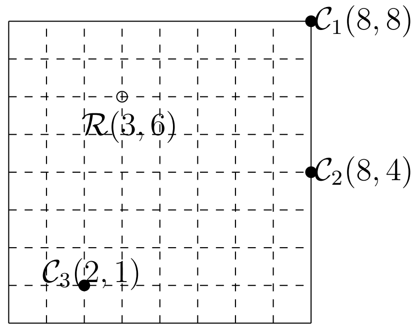


Figure 2.1.1: $\mathcal{C}_1$, $\mathcal{C}_2$ and $\mathcal{C}_3$ are in corner, boundary and internal vertex respectively. Here $h = 1$ and $v = 2$.

grid and all the internal faces have unit area.

Now consider the grid $P_m \Box P_n$. Let the cop $\mathcal{C}_i$ be at vertex (x_i, y_i) and the robber $\mathcal{R}$ be at a vertex (x_r, y_r). We define $row(\mathcal{C}_i) = y_i$ and $row(\mathcal{R}) = y_r$. Moreover, we define $h(\mathcal{C}_i)$ and $v(\mathcal{C}_i)$ as $h(\mathcal{C}_i) = |x_i - x_r|$ and $v(\mathcal{C}_i) = |y_i - y_r|$. Let $h = \min_{\mathcal{C}_i} h(\mathcal{C}_i)$ and $v = \min_{\mathcal{C}_i} v(\mathcal{C}_i)$ (see Fig. 2.1.1). We say $\mathcal{C}_i$ *moves towards* $\mathcal{R}$ if either $h(\mathcal{C}_i)$ or $v(\mathcal{C}_i)$ decreases. Similarly, we say that $\mathcal{R}$ *moves away from* $\mathcal{C}_i$ if either $h(\mathcal{C}_i)$ or $v(\mathcal{C}_i)$ increases. We also say that $\mathcal{C}_i$ moves vertically (or horizontally) towards $\mathcal{R}$ if $h(\mathcal{C}_i)$ (or $v(\mathcal{C}_i)$) decreases, and $\mathcal{R}$ moves vertically (or horizontally) away from $\mathcal{C}_i$ if $h(\mathcal{C}_i)$ (or $v(\mathcal{C}_i)$) increases. Cop $\mathcal{C}_i$ at (x_i, y_i) moves towards a vertex (p, q) if $|x_i - p|$ or $|y_i - q|$ decreases.

Many of our strategies are described in terms of rounds, that is, they say how the cops will move depending on the robber's move. Therefore, to avoid confusion, unless mentioned otherwise, we fix the following. After the cops are placed, the first round begins with the placement of the robber. In a round, the cop $\mathcal{C}_i$ *mirrors* the move of $\mathcal{R}$ if $h(\mathcal{C}_i)$ and $v(\mathcal{C}_i)$ do not change after that round, that is, in this round $\mathcal{C}_i$ moves exactly in the same direction as $\mathcal{R}$. We say that cops $\mathcal{C}_i, \ldots, \mathcal{C}_k$ *regroup* at a vertex v if all of them reach v in a finite number of rounds.

In $P_m \Box P_n$, $C_m \Box P_n$ and $C_m \Box C_n$, we say that $\mathcal{C}_i$ *blocks* $\mathcal{R}$ if $h(\mathcal{C}_i) = 1$

and $v(\mathcal{C}_i) = 1$. Suppose $\mathcal{C}_i$ blocks $\mathcal{R}$ in $P_m \Box P_n$ and $\mathcal{C}_i$ is not at a corner vertex. If we draw the co-ordinate axes with origin at $\mathcal{C}_i$, it divides the grid in four quadrants each containing one corner point. In $P_m \Box P_n$, $\mathcal{R}$ is *trapped* by $\mathcal{C}_i$ when $\mathcal{R}$ is in a corner vertex and $\mathcal{C}_i$ blocks $\mathcal{R}$. When $\mathcal{R}$ is trapped, it cannot make any move without getting captured.

We say that cop/cops guard a subgraph H of G if $\mathcal{R}$ cannot enter the vertices of H without getting captured in the next cop move. In the context of grids, we guard rows/columns to capture the robber. Observe that if a cop C is in the row $y = j$ and $h(\mathcal{C}) \leq 1$, then $\mathcal{R}$ cannot enter the row $y = j$ without getting captured. Similarly, if C is in the column $x = i$ and $v(\mathcal{C}) \leq 1$, then $\mathcal{R}$ cannot enter the the column $x = i$ without getting captured. Hence, we say that a cop $\mathcal{C}$ *guards* the row $y = j$ if $\mathcal{C}$ is in $y = j$ and $h(\mathcal{C}) \leq 1$. Similarly, $\mathcal{C}$ *guards* the column $x = i$ if $\mathcal{C}$ is in $x = i$ and $v(\mathcal{C}) \leq 1$. Once a cop guards a row/column, it can keep guarding that row/column by mirroring the robber's move whenever it loses the guarding position.

Consider the classical Cops and Robber game on a planar grid. Here one cop can always guard a row/column. We have the following observation.

Observation 2.1.1. *In the classical Cops and Robber game, one cop can guard a row/column of a planar grid.*

Proof. We will show how to guard a row $y = j$. If the robber is at vertex (i', j'), then we consider the *image* of the robber at vertex (i', j). The image of the robber moves as the robber moves but is restricted to a finite path (induced by the row $y = j$), and the image can only move to a neighbouring vertex. The cop will move on the vertices of this path and will capture the image of the robber. Observe that at this point the cop guards the row $y = j$. The cop can similarly guard a column $x = i$. $\Box$

Next, we show that in the game of classical Cops and Robber on a planar grid, one cop can block the robber. We have the following lemma which will be useful for several results in this thesis.

Lemma 2.1.1. *In the classical Cops and Robber game on a planar grid, one cop can block the robber.*

Proof. The cop $\mathcal{C}$ begins at vertex $(0, 0)$. Now $\mathcal{C}$ will move depending on the moves of $\mathcal{R}$ in the following manner until $\mathcal{C}$ blocks $\mathcal{R}$.

1. If $\mathcal{R}$ moves such that either $h(\mathcal{C})$ or $v(\mathcal{C})$ increases, then $\mathcal{C}$ mirrors the move of $\mathcal{R}$.

2. If $\mathcal{R}$ move such that neither $h(\mathcal{C})$ nor $v(\mathcal{C})$ increases, then $\mathcal{C}$ moves such that $\max\{h(\mathcal{C}), v(\mathcal{C})\}$ decreases.

Next, we show that $\mathcal{R}$ cannot make moves of type 1 forever. Without loss of generality, let us assume that $\mathcal{C}$ is at a vertex (i, j) and $\mathcal{R}$ is at a vertex $(i + a, j + b)$, where $a \geq 0$ and $b \geq 0$. Now to increase either $h(\mathcal{C})$ or $v(\mathcal{C})$, $\mathcal{R}$ has to make either a horizontal right move or a vertical up move. Since the grid is finite, $\mathcal{R}$ cannot make these moves forever and $\mathcal{R}$ has to make a move of type 2 after every finite number of rounds. Thus $\max\{h(\mathcal{C}), v(\mathcal{C})\}$ decreases after every finite number of rounds.

Hence, after a finite number of rounds, both $h(\mathcal{C})$ and $v(\mathcal{C})$ become equal to 1, and $\mathcal{C}$ blocks $\mathcal{R}$. □

2.2 Chapter overview

In Section 2.3, we consider the game of attacking Cops and Robber on various kinds of grids originating from the Cartesian product of paths and cycles. We show that the attacking cop number of planar grids is 2. Then we show that the attacking cop number of both cylindrical grids and toroidal grids is 3. Next, we consider the game of Cops and attacking Robber on hypercubes and show that in a d-dimensional hypercube $\lceil \frac{d}{2} \rceil + 1$ cops are sufficient to capture the attacking robber. We also show that for a d-dimensional grid, d cops are sufficient to capture the attacking robber. Recall that the cop number of d-dimensional hypercube and d-dimensional grid for the classical Cops and Robber game is $\lceil \frac{d+1}{2} \rceil$ [80].

In Section 2.4, we present our results on lazy cops and robbers. We first show that the lazy cop number for both planar grids and cylindrical grids is 2. It seems that on toroidal grids ($C_m \square C_n$) a *few* lazy cops would not create problems for the robber. One of the ways to strengthen the ability of cops is to add a few *flexible* cops. In such a setting, all of the flexible cops are free to move in every round, while only one of the lazy cops can move in a round. We show that one flexible and two lazy cops can capture the robber on toroidal grids.

In Section 2.5, we consider the game of classical Cops and Robber on the subgraph of planar grids. We show that the cop number for the class of solid grids is 2. Then we show that the cop number for the class of partial grids is 3.

Finally, we draw conclusions in Section 2.6.

2.3 Cops and attacking Robber

In this section, we consider the game of Cops and attacking Robber on various kinds of grids. We give strategies to capture the attacking robber on planar grids, cylindrical grids, toroidal grids, hypercubes, and d-dimensional grids.

2.3.1 Cops and attacking Robber on Planar grids

In this section, we prove that the attacking cop number of the planar grids is 2. We have the following theorem.

Theorem 2.3.1. *The attacking cop number of a finite planar grid is 2, that is, $cc(P_m \square P_n) = 2$.*

Proof. Neufeld and Nowakowski [83] proved that $c(P_m \Box P_n) = 2$. Since $c(G) \leq cc(G)$, the attacking cop number is at least 2. Thus, it suffices to give a strategy to capture the attacking robber using two cops to show that $cc(P_m \Box P_n) = 2$.

Let the two cops be $\mathcal{C}_1$ and $\mathcal{C}_2$. To avoid getting attacked by the robber, both cops will either remain on the same vertex or adjacent vertices at all points in the game. The cops will use the following strategy.

Both cops start at the vertex $(0, 0)$. Then $\mathcal{C}_1$ blocks $\mathcal{R}$ using Lemma 2.1.1 and $\mathcal{C}_2$ moves such that after each cop move, both $\mathcal{C}_1$ and $\mathcal{C}_2$ are on the same vertex. Without loss of generality, let us assume that both cops are at vertex (i, j) and $\mathcal{R}$ is at vertex $(i + 1, j + 1)$. Now $\mathcal{R}$ can only make either a vertical up move or a horizontal right move. If $\mathcal{R}$ makes such a move, both the cops mirror the move of $\mathcal{R}$ and retain their block position. Since the grid is finite, eventually $\mathcal{R}$ will end up at the top-right corner vertex $(m-1, n-1)$ if $\mathcal{R}$ keeps moving. If at some point in the game, $\mathcal{R}$ is at vertex (x, y) (cops are at vertex $(x-1, y-1)$) and $\mathcal{R}$ does not move on a robber move, then $\mathcal{C}_2$ moves to vertex $(x - 1, y)$. Now the robber is forced to move and both cops will mirror the moves of $\mathcal{R}$. Hence, $\mathcal{R}$ eventually ends up at vertex $(m - 1, n - 1)$.

Now $\mathcal{R}$ is at vertex $(m - 1, n - 1)$, and either both cops are on vertex $(m - 2, n - 2)$ or $\mathcal{C}_1$ is at vertex $(m - 2, n - 2)$ and $\mathcal{C}_2$ is at vertex $(m - 2, n - 1)$. In any case, in the next cop move, both cops move such that $\mathcal{C}_1$ is at vertex $(m - 2, n - 2)$ and $\mathcal{C}_2$ is at vertex $(m - 2, n - 1)$. In the next round, irrespective of the moves of $\mathcal{R}$, $\mathcal{R}$ will be captured by the cops. $\square$

2.3.2 Cops and attacking Robber on cylindrical grids

In this section, we prove that the attacking cop number of the cylindrical grids is 3, that is, $cc(C_m \Box P_n) = 3$. We have the following theorem.

Theorem 2.3.2. *The attacking cop number of a finite cylindrical grid $C_m \square P_n$ (for $m > 3$) is 3, that is, $cc(C_m \square P_n) = 3$.*

Proof. Luccio and Pagli [79] proved that $c(C_m \square P_n) = 2$. Therefore, to prove $cc(C_m \square P_n) = 3$, we will give both a strategy to capture the attacking robber using three cops and an example of a cylindrical grid where three cops are necessary to capture the attacking robber. Let the three cops be $\mathcal{C}_1, \mathcal{C}_2$, and $\mathcal{C}_3$

Strategy Outline

1. Cop $\mathcal{C}_3$ will guard the column $x = 0$.
2. Robber $\mathcal{R}$ is restricted to a planar grid $P_{m-1} \square P_n$, and then $\mathcal{C}_1$ and $\mathcal{C}_2$ capture $\mathcal{R}$ using the strategy from Theorem 2.3.1.

Initially all cops $\mathcal{C}_1$, $\mathcal{C}_2$, and $\mathcal{C}_3$ start at $(0, 0)$. Cops $\mathcal{C}_3$ and $\mathcal{C}_1$ move together (to avoid getting attacked by $\mathcal{R}$) and guard the column $x = 0$ using Observation 2.1.1. Once $\mathcal{C}_3$ guards $x = 0$, observe that an attacking robber cannot enter $x = 0$. Now $\mathcal{C}_1$ moves in column $x = 0$ and regroups with $\mathcal{C}_2$ at $(0, 0)$. Note that, during this strategy, when a cop is alone, it cannot be attacked by $\mathcal{R}$.

Since $\mathcal{R}$ cannot enter column $x = 0$ (as it is guarded by $\mathcal{C}_3$), $\mathcal{R}$ is restricted to move in a planar grid $P_{m-1} \square P_n$. Now cops $\mathcal{C}_1$ and $\mathcal{C}_2$ capture $\mathcal{R}$, by using strategy from Theorem 2.3.1.

Now we prove that three cops are necessary for cylindrical grids. It can be verified that to capture an attacking robber in $C_m \square P_n$, for $m > 3$, at least three cops are necessary. In the last round, before two cops capture the attacking robber, they should be adjacent to each other and adjacent to $\mathcal{R}$ and its neighbors. Since no such position of two cops and an attacking robber exists in $C_m \square P_n$, two cops cannot capture an attacking robber. $\square$

However, we have the following remark about the attacking cop number of $C_3 \square P_n$.

Remark 2.3.1. *Note that in $C_3 \square P_n$, two cops can capture an attacking robber.*

2.3.3 Cops and attacking Robber on Toroidal Grids

In this section, we prove that the attacking cop number of the toroidal grids is 3. We have the following theorem.

Theorem 2.3.3. *The attacking cop number of a toroidal grid is 3, that is, $cc(C_m \square C_n) = 3$.*

Proof. Neufeld and Nowakowski [83] proved that $c(C_m \square C_n) = 3$. Since $c(G) \leq cc(G)$, $cc(C_m \square C_n) \geq 3$. Hence, it suffices to give a strategy to capture an attacking robber using three cops, say $\mathcal{C}_1$, $\mathcal{C}_2$ and $\mathcal{C}_3$.

Strategy Outline

1. All cops start at $(0, 0)$.
2. $\mathcal{C}_1$ guards the column $x = 0$.
3. $\mathcal{C}_2$ and $\mathcal{C}_3$ guard the row $y = 0$.
4. $\mathcal{C}_3$ blocks the robber.
5. Cops now capture the robber.

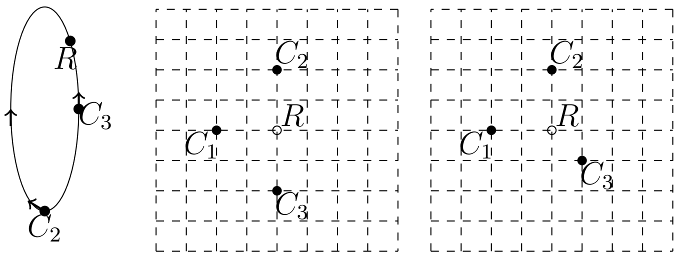
Now we give the detailed strategy. One of the cop, say $\mathcal{C}_1$, remains at $(0, 0)$ while $\mathcal{C}_2$ and $\mathcal{C}_3$ move to guard the column $x = 0$. If the robber $\mathcal{R}$ ever enters the row $y = 0$, the column $x = 0$ gets guarded by $\mathcal{C}_1$. If $\mathcal{R}$ never enters the row $y = 0$, its movements are restricted in a cylindrical grid $C_m \square P_{n-1}$, with each row inducing a cycle C_m. Now using the strategy similar to the first step in the proof of Theorem 2.3.2, $\mathcal{C}_2$ and $\mathcal{C}_3$ guard the column $x = 0$. Hence, after a finite number of rounds, either a cop is guarding a row or a cop is guarding a column. Now rename the cops such that $\mathcal{C}_1$ is guarding either the column $x = 0$ or the row $y = 0$. Without loss of generality, let us assume that $\mathcal{C}_1$ is guarding the column $x = 0$.

After this, $\mathcal{R}$ cannot enter the column $x = 0$ without being captured. Hence, $\mathcal{R}$ is restricted to a cylindrical grid $P_{m-1}\Box C_n$, with each column inducing a cycle C_n.

Now cops use a strategy similar to the first step in the proof of Theorem 2.3.2: $\mathcal{C}_2$ and $\mathcal{C}_3$ guard the row $y = 0$ (we reconfigure the axes system if required by renaming the guarded row $y = 0$ and accordingly renaming the other rows). Now $\mathcal{C}_1$ is guarding the column $x = 0$ and $\mathcal{C}_2$ and $\mathcal{C}_3$ are guarding the row $y = 0$. Hence, $\mathcal{R}$ is restricted to a planar grid $P_{m-1}\Box P_{n-1}$. At this point, we assume that $v(\mathcal{C}_1) = 0$, $h(\mathcal{C}_2) = 0$ and $h(\mathcal{C}_3) = 0$ (this can be attained in finite steps).

Now $\mathcal{C}_3$ moves to block $\mathcal{R}$. At each step, $v(\mathcal{C}_1) = 0$, $h(\mathcal{C}_2) = 0$ and $h(\mathcal{C}_3) \leq 1$. After finite rounds, either $\mathcal{C}_3$ blocks $\mathcal{R}$ or $h(\mathcal{C}_3) = 0$ and $v(\mathcal{C}_3) = 2$. In the latter case, if $\mathcal{R}$ ever moves horizontally, then $\mathcal{C}_3$ moves to block $\mathcal{R}$. If $\mathcal{R}$ does not move or moves vertically, then $\mathcal{C}_2$ and $\mathcal{C}_3$ approach $\mathcal{R}$ from opposite directions (see Figure 2.3.1(a)) with $\mathcal{C}_3$ maintaining $v(\mathcal{C}_3) = 2$. After a finite number of rounds, we have $v(\mathcal{C}_1) = 0$, $h(\mathcal{C}_3) = 0$, $v(\mathcal{C}_3) = 2$, $h(\mathcal{C}_2) = 0$, $v(\mathcal{C}_2) = 2$. In this position, if $\mathcal{R}$ moves vertically, then it gets captured, and if $\mathcal{R}$ moves horizontally, then $\mathcal{C}_3$ moves to block $\mathcal{R}$. If $\mathcal{R}$ chooses not to move in this situation, then $\mathcal{C}_1$ moves toward $\mathcal{R}$ till $h(\mathcal{C}_1) = 2$ (see Figure 2.3.1(b)). Now $\mathcal{C}_3$ moves right; $\mathcal{R}$ cannot move left or up. If $\mathcal{R}$ moves down or stays put, then $\mathcal{C}_3$ blocks $\mathcal{R}$. If $\mathcal{R}$ moves right, then $\mathcal{C}_2$ blocks $\mathcal{R}$ (without loss of generality, assume that $\mathcal{C}_3$ blocks $\mathcal{R}$: see Figure 2.3.1(c)). Now we have $\mathcal{C}_1$ and $\mathcal{C}_2$ guarding a column and a row, respectively with $h(\mathcal{C}_1) = 2,\ v(\mathcal{C}_1) = 0,\ h(\mathcal{C}_2) = 0,\ v(\mathcal{C}_2) = 2,\ h(\mathcal{C}_3) = 1,\ v(\mathcal{C}_3) = 1$.

Now $\mathcal{C}_1$ moves up. If $\mathcal{R}$ moves left or stays put, then we end up with $\mathcal{C}_1$ and $\mathcal{C}_3$ blocking $\mathcal{R}$ with $h(\mathcal{C}_1, \mathcal{C}_3) = 2$, $v(\mathcal{C}_1, \mathcal{C}_3) = 2$, $v(\mathcal{C}_2) = 2$ and $h(\mathcal{C}_2) = 1$. Now $\mathcal{C}_2$ moves down. Now whatever $\mathcal{R}$ does, it gets captured in the next cop move. This completes the proof of Theorem 2.3.3. $\square$



(a) (b) (c)

Figure 2.3.1: Attacking robber on toroidal grid

2.3.4 Cops and attacking Robber on multidimensional grids

In this section, we study the game of Cops and attacking Robber on higher-dimensional grids. We show that for a d-dimensional grid, the attacking cop number is at most d, that is, $cc(P_{i_1}\square P_{i_2}\square \dots \square P_{i_d}) \leq d$. We have the following theorem.

Theorem 2.3.4. *The attacking cop number of a d-dimensional grid is at most d, that is,* $\lceil \frac{d+1}{2} \rceil \leq cc(P_{i_1}\square P_{i_2}\square \dots \square P_{i_d}) \leq d$.

Proof. The lower bound is obvious as the classical cop number of d-dimensional grids is $\lceil \frac{d+1}{2} \rceil$ [80]. To prove the upper bound, we use induction on d. The base case, $cc(P_m \square P_n) \leq 2$, follows from Theorem 2.3.1. We assume that for some $d \in \mathbb{N}$, $cc(P_{i_1}\square P_{i_2}\square \cdots \square\ P_{i_{d-1}}) \leq d-1$. In what follows, we give a strategy for d cops to capture an attacking robber in $P_{i_1}\square P_{i_2}\square \cdots \square P_{i_d}$.

For the sake of convenience, let A represent the $(d-1)$-dimensional grid $P_{i_1}\square P_{i_2}\square \cdots \square P_{i_{d-1}}$. Hence $P_{i_1}\square P_{i_2}\square \cdots \square P_{i_d}$ can be considered as i_d copies of A with corresponding vertices between A_i and A_{i+1} being adjacent (see Fig. 2.3.2). By *image* I_c of the robber in A_c, we mean the vertex in A_c which differs from the robber's position only in the d^{th} coordinate. We say that a cop $\mathcal{C}_i$ *captures the image* I_c if $\mathcal{C}_i$ and I_c are on the same vertex after the cop's move. We say that a cop $\mathcal{C}_i$ *protects the image* I_c if $\mathcal{R}$ is not in A_c and $\mathcal{C}_i$ is adjacent to I_c after the cop's move.

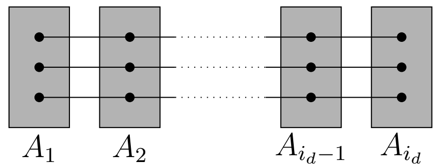


Figure 2.3.2: d-dimensional grid partitioned into $d-1$-dimensional grid

Observe that, if a cop is protecting I_c, then $\mathcal{R}$ cannot enter A_c without getting captured.

Strategy Outline:

1. All cops start in A_1 and capture I_1.
2. Cop $\mathcal{C}_1$ protects or captures I_1. Other $d-1$ cops regroup and move to A_2.
3. For $x = 2$ to $k-1$:
   (a) Cop, say $\mathcal{C}_2$, captures I_x.
   (b) Rest $d-1$ cops regroup in A_x while $\mathcal{C}_2$ guards I_x.
   (c) Rename $\mathcal{C}_2$ as $\mathcal{C}_1$ and $\mathcal{C}_1$ as $\mathcal{C}_2$.
   (d) The $d-1$ cops, except $\mathcal{C}_1$, move to A_{x+1}.
4. Capture $\mathcal{R}$ in A_{i_d}.

All d cops start in A_1. By our induction hypothesis, $d-1$ of them, except say $\mathcal{C}_2$, captures I_1. During this time, $\mathcal{C}_2$ moves with a cop so that $\mathcal{R}$ cannot eliminate it. Let $\mathcal{C}_1$ be the cop that captures I_1. Now $\mathcal{R}$ makes its move. Depending on $\mathcal{R}$'s move, $\mathcal{C}_1$ can move (or stay put) such that it is protecting I_1. Thus if $\mathcal{R}$ enters A_1, then $\mathcal{C}_1$ will capture $\mathcal{R}$. All other $d-1$ cops regroup at a vertex in A_1 and then move to A_2, while $\mathcal{C}_1$ keeps on protecting I_1 by mirroring $\mathcal{R}$'s move. Next, the $d-1$ cops in A_2 capture I_2. Say $\mathcal{C}_2$ protects I_2, and the rest regroup at a vertex in A_2. Iteratively, we can proceed such that some cop is protecting I_{d-1}

and the other $d-1$ cops and $\mathcal{R}$ are in A_{i_d}, where $\mathcal{R}$ gets captured by our induction hypothesis.

This completes the proof of Theorem 2.3.4. $\square$

2.3.5 Cops and attacking Robber on Hypercubes

In this section, we consider the game of Cops and attacking Robber on hypercubes. We prove that the attacking cop number of the d- dimensional hypercube is at most $\lceil \frac{d}{2} \rceil + 1$. We have the following theorem.

Theorem 2.3.5. *The attacking cop number of the d-dimensional hypercube is at most* $\lceil \frac{d}{2} \rceil + 1$, *that is,* $\lceil \frac{d+1}{2} \rceil \leq cc(Q_d) \leq \lceil \frac{d}{2} \rceil + 1$.

Proof. The lower bound is obvious as the classical cop number of d-dimensional grids is $\lceil \frac{d+1}{2} \rceil$ [80]. To prove the upper bound, we use induction on the dimension of the hypercube. Since $Q_{d+1} = Q_d \Box K_2$, it can be considered as two copies of Q_d with edges between corresponding vertices. Similarly, Q_{d+2} can be considered as four copies of Q_d, say A, B, C and D (see Figure 2.3.3) with corresponding edges between $(A, B), (B, C), (C, D)$ and (D, A). If the robber $\mathcal{R}$ is in one of the Q_d, say in C, then the vertex in B and D adjacent to $\mathcal{R}$ are denoted I_B and I_D respectively. The vertex in A adjacent to I_B and I_D is denoted by I_A. The vertices I_A, I_B and I_D are called the *image* of the robber (see Figure 2.3.3). If a cop is adjacent to I_K, where K denotes some hypercube Q_d, we say that it is *protecting* I_K and also K is guarded by the cop. Moreover, if a cop is on vertex I_K, we say it has captured I_K.

For the base case of induction, $cc(Q_2) \leq 2$ is obvious and $cc(Q_3) \leq 3$ is a special case of Theorem 2.3.4 as $Q_3 = P_2 \Box P_2 \Box P_2$. We assume that for some $d \in N$, $cc(Q_d) \leq \lceil \frac{d}{2} \rceil + 1$. In what follows, we give a strategy for $\lceil \frac{d}{2} \rceil + 2$ cops to capture an attacking robber in Q_{d+2}.

Strategy Outline

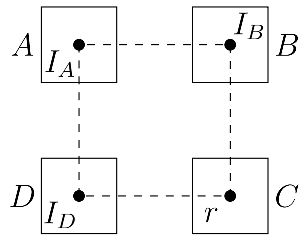


Figure 2.3.3: Hypercube Q_{d+2} decomposed into four Q_d's.

1. All cops start in A.
2. Cops capture/protect I_A.
3. Cops force $\mathcal{R}$ to move to C and then restrict it to C.
4. Cops capture $\mathcal{R}$ in C.

All cops start in A. The extra cop, that is, $cc(Q_d) + 1^{\text{th}}$ cop starts and moves with one of the cops so that it does not get eliminated by the attacking robber. By our induction hypothesis, $cc(Q_d)$ cops capture I_A in A. When a cop captures the attacking robber or its image, it is adjacent to another cop, else the robber could have eliminated it. Let $\mathcal{C}_1$ be the cop capturing I_A and $\mathcal{C}_2$ be adjacent to $\mathcal{C}_1$. So $\mathcal{C}_2$ is protecting I_A, and hence $\mathcal{R}$ cannot enter A.

If $\mathcal{R}$ is in C, then $\mathcal{C}_1$ maintains the capture position. So $\mathcal{R}$ cannot enter B or D. All other cops regroup in A and then move to C via B or D. By our induction hypothesis, $cc(Q_d)$ cops capture $\mathcal{R}$ in C.

If $\mathcal{R}$ is in B or D (without loss of generality assume that $\mathcal{R}$ is in B), then $\mathcal{C}_2$ keeps on protecting I_A so that $\mathcal{R}$ cannot enter A. Then rest of the cops regroup in A and then move to B. By our induction hypothesis, $cc(Q_d)$ cops can capture $\mathcal{R}$ in B if $\mathcal{R}$ does not move to C; else they capture I_B. In the round when $\mathcal{R}$ moves from B to C, $\mathcal{C}_2$ captures I_A (as I_A does not change in this round and $\mathcal{C}_2$ was protecting I_A). Thus $\mathcal{R}$ cannot enter B or D. All other cops regroup in B and then move to C. By our induction hypothesis, $cc(Q_d)$ cops capture $\mathcal{R}$ in C. $\square$

2.4 Lazy cops and robber

In this section, we consider the game of lazy Cops and Robber on grids. In particular, we give bounds for the lazy cop number of planar and cylindrical grids.

2.4.1 Lazy Cops and Robber on Planar Grids

In this section, we consider the game of lazy Cops and Robber on planar grids. We prove that the lazy cop number for the planar grids is 2. We have the following theorem.

Theorem 2.4.1. *The lazy cop number for planar grids is 2, that is,* $lc(P_m \square P_n) = 2$.

Proof. Clearly $c(G) \leq lc(G)$ as $lc(G)$ number of cops can capture the robber in the classical cops and robbers game. And since the cop number of a planar grid $(P_m \square P_n)$ is 2 (as proved by Neufeld and Nowakowski [83]), $lc(P_m \square P_n) \geq 2$. Hence it suffices to give a strategy to capture the robber using two lazy cops $\mathcal{C}_1$ and $\mathcal{C}_2$ to prove $lc(P_m \square P_n) = 2$.

We begin with an outline of such a strategy. If a lazy cop $\mathcal{C}_1$ traps the robber $\mathcal{R}$, then a lazy cop $\mathcal{C}_2$ can capture $\mathcal{R}$ in finite time. Let $\mathcal{C}_1$ and $\mathcal{C}_2$ both be initially placed at $(0, 0)$. First $\mathcal{C}_1$ blocks $\mathcal{R}$. In further rounds, $\mathcal{C}_1$ and $\mathcal{C}_2$ will force $\mathcal{R}$ to go in a trap position such that $\mathcal{R}$ is trapped by $\mathcal{C}_1$. Then $\mathcal{C}_2$ moves towards $\mathcal{R}$ till it captures $\mathcal{R}$. We use the following simple observation.

Observation 2.4.1. *If robber $\mathcal{R}$ gets into trap position of a cop $\mathcal{C}_1$, $\mathcal{R}$ cannot move without being captured.*

Now we give the detailed strategy. Only one of the lazy cops can move in a cop's turn. First, cop $\mathcal{C}_1$ blocks $\mathcal{R}$ using Lemma 2.1.1. In all these rounds, only the cop $\mathcal{C}_1$ moves. Without loss of generality, we assume that $\mathcal{C}_1$ blocks $\mathcal{R}$ such that $\mathcal{C}_1$ is at vertex (x, y) and $\mathcal{R}$ is at vertex $(x+1, y+1)$.

Now, the only moves $\mathcal{R}$ can make are horizontally right and vertically up. Whenever $\mathcal{R}$ moves, $\mathcal{C}_1$ mirrors the move of $\mathcal{R}$, thus maintaining its block position. Moreover, if $\mathcal{R}$ keeps moving, then eventually $\mathcal{R}$ will end up at the corner vertex $(m-1, n-1)$ with $\mathcal{C}_1$ at vertex $(m-2, n-2)$.

If $\mathcal{R}$ chooses not to move, then $\mathcal{C}_2$ moves towards $\mathcal{R}$ and will eventually force $\mathcal{R}$ to move. Hence, eventually $\mathcal{R}$ will end up at the $(m-1, n-1)$ with $\mathcal{C}_1$ at vertex $(m-2, n-2)$. Now $\mathcal{R}$ cannot make a move, and $\mathcal{C}_2$ will move and capture $\mathcal{R}$. □

2.4.2 Lazy Cops and Robber on Cylindrical Grids

In this section, we consider the game of lazy Cops and Robber on cylindrical grids and prove that the lazy cop number of cylindrical grids is 2. We have the following theorem.

Theorem 2.4.2. *The lazy cop number of cylindrical grids is 2, that is,* $lc(C_m \square P_n) = 2$.

Proof. Luccio and Pagli [79] proved that $c(C_m \square P_n) = 2$. Thus, it suffices to give a strategy to capture the robber $\mathcal{R}$ using two lazy cops $\mathcal{C}_1$ and $\mathcal{C}_2$ to prove $lc(C_m \square P_n) = 2$.

Strategy Outline

1. Both cops start at $(0, 0)$.
2. $\mathcal{C}_1$ guards the column $x = 0$.
3. $\mathcal{C}_2$ makes $h(\mathcal{C}_2) = 0$.
4. $\mathcal{C}_1$ and $\mathcal{C}_2$ restricts $\mathcal{R}$ to one of boundary rows.
5. $\mathcal{R}$ is captured.

All cops start at $(0, 0)$. $\mathcal{C}_1$ guards the column $x = 0$ using Observation 2.1.1. Now $\mathcal{C}_2$ moves to make $h(\mathcal{C}_2) = 0$. This can be done as following. In each round, $\mathcal{R}$ can move in one of the following ways:

- $\mathcal{R}$ moves vertically: In this case, if $\mathcal{R}$ makes $v(\mathcal{C}_1) > 1$, $\mathcal{C}_1$ follows $\mathcal{R}$ hence ensuring column $x = 0$ is always guarded by $\mathcal{C}_1$. On any other move of $\mathcal{R}$, $\mathcal{C}_2$ moves to decrease $h(\mathcal{C}_2)$.

- $\mathcal{R}$ moves horizontally or does not move: $\mathcal{C}_2$ moves trying to decrease $h(\mathcal{C}_2)$.

Since $\mathcal{C}_1$ always guards column $x = 0$, $v(\mathcal{C}_1) \leq 1$. Moreover, the value of $h(\mathcal{C}_2)$ never increases after a round, because if $\mathcal{R}$ moves horizontally away from $\mathcal{C}_2$, then $\mathcal{C}_2$ mirrors $\mathcal{R}$'s move. Since column $x = 0$ is guarded and the number of columns is finite, $\mathcal{R}$ moves horizontally away from $\mathcal{C}_2$ for a finite rounds, after which $h(\mathcal{C}_2)$ decreases. After a finite number of rounds $h(\mathcal{C}_2)$ decreases to 0. Now we have $h(\mathcal{C}_2) = 0$ and $v(\mathcal{C}_1) \leq 1$.

Now $\mathcal{C}_1$ and $\mathcal{C}_2$ will try to restrict $\mathcal{R}$ to one of the *boundary rows*, that is, the topmost or the bottommost row. Without loss of generality, let us assume that $row(\mathcal{C}_1) \geq row(\mathcal{C}_2)$ and $row(\mathcal{R}) > row(\mathcal{C}_2)$. If $\mathcal{R}$ moves to make $h(\mathcal{C}_2) > 1$ or $v(\mathcal{C}_1) > 1$, then $\mathcal{C}_2$ or $\mathcal{C}_1$ follows $\mathcal{R}$ respectively. On any other move of $\mathcal{R}$, $\mathcal{C}_2$ moves vertically up towards $\mathcal{R}$. Here we ensure that $h(\mathcal{C}_2) \leq 1$ and $v(\mathcal{C}_1) \leq 1$ is maintained, that is, $\mathcal{C}_1$ guards $x = 0$ and $\mathcal{C}_2$ guards its row. Therefore $\mathcal{R}$ is always restricted to the rows above $\mathcal{C}_2$. Since $\mathcal{C}_2$ never moves down vertically, but moves up vertically after every finite number of moves, $\mathcal{R}$ will eventually be restricted only to the topmost row. We can also ensure that $h(\mathcal{C}_2) = 1$ and $v(\mathcal{C}_2) = 1$. Now $\mathcal{R}$ can only make horizontal moves in one direction. Whenever $\mathcal{R}$ makes any horizontal move, $\mathcal{C}_2$ mirrors $\mathcal{R}$'s move and maintains the block position. In the meantime, if $\mathcal{R}$ chooses not to move, $\mathcal{C}_1$ moves horizontally in the opposite direction to the direction $\mathcal{R}$ is allowed to move. Thus after a finite number of moves, $\mathcal{R}$ will be restricted to the topmost row with $\mathcal{C}_1$ and $\mathcal{C}_2$ blocking $\mathcal{R}$ from both sides (left and right of $\mathcal{R}$).

Now $\mathcal{R}$ cannot move and will be captured in at most two rounds. $\square$

2.4.3 Lazy and Flexible Cops and Robber on Toroidal Grids

One of the ways to strengthen the ability of lazy cops is to add a few *flexible* cops. In such a setting, all of the *flexible* cops are free to move in every round while only one of the lazy cops can move in a round. Neufeld and Nowakowski [83] proved that the cop number of toroidal grids is three. So three $flexible$ cops can capture the robber in toroidal grids. In what follows, we prove that one flexible cop and two lazy cops can capture a robber on toroidal grids.

During a cop's turn, a flexible cop can move or pass its turn whereas at most one of the lazy cops can move. Let $\mathcal{C}_1$ be the flexible cop, and $\mathcal{C}_2$, $\mathcal{C}_3$ be the lazy cops. We now give a strategy for the cops to capture $\mathcal{R}$ on toroidal grid $C_m \square C_n$. We have the following theorem.

Theorem 2.4.3. *One flexible and two lazy cops can capture the robber in a finite toroidal grid.*

Proof. We give a cop strategy to prove our theorem.

Strategy Outline

1. All cops start at $(0, 0)$.
2. A lazy cop guards the column $x = 0$.
3. $\mathcal{C}_1$ guards the row $y = 0$.
4. $\mathcal{C}_2$ and $\mathcal{C}_3$ capture $\mathcal{R}$ using the strategy used in the proof of Theorem 2.4.2.

All cops start at $(0, 0)$. Cop $\mathcal{C}_2$ moves to guard the column $x = 0$. If $\mathcal{R}$ ever enters the row $y = 0$, then the column $x = 0$ gets guarded by $\mathcal{C}_1$ and $\mathcal{C}_3$. If $\mathcal{R}$ never enters $x = 0$, then $\mathcal{C}_2$ can guard the column $x = 0$ as movements of $\mathcal{R}$ are restricted to a cylindrical grid $C_m \square P_{n-1}$. Thus after a finite number of moves, the column $x = 0$ is guarded by one of the lazy cops. Now $\mathcal{C}_1$ can guard the row $y = 0$ as $\mathcal{R}$ cannot enter the column $x = 0$ and is restricted to a cylindrical grid $C_n \square P_{m-1}$.

Now $\mathcal{C}_1$ is guarding the row $x = 0$. Hence, the robber is restricted to a cylindrical grid $C_m \square P_{m-1}$. Now lazy cops $\mathcal{C}_3$ and $\mathcal{C}_2$ use the strategy used in the proof of Theorem 2.4.2 to capture $\mathcal{R}$. $\square$

2.5 Classical Cops and Robber on subgraphs of Grids

In this section, we consider the game of classical Cops and Robber on the subgraphs of planar grids, that is, solid grids and partial grids. We provide a dichotomy result proving that while 2 cops can always ensure capture of the robber for solid grids, there are partial grids where 3 cops are necessary to capture the robber.

2.5.1 Cops and Robber on solid grids

In this section, we consider the game of classical Cops and Robber on solid grids. We consider a grid representation of the solid grid graph. In this representation rows and columns are clearly defined. For sake of simplicity, we assume that our solid grid has more than one columns.

A *column path* is a path which has all its vertices from the same column, say c_i, and both endpoints of this path have exactly one neighbour in c_i, each. A column c_i may have multiple column paths. A column path P in column c_i is a *boundary column path* if vertices of P have neighbours only in P and in either column c_{i+1} or in c_{i-1}. See Figure 2.5.1 for an illustration. Two column paths P and P' are *adjacent* if some vertex $p \in P$ and $p' \in P'$ have an edge.

Let P be a column path of a solid grid graph G, and P have endpoints u and v. It is easy to see that:

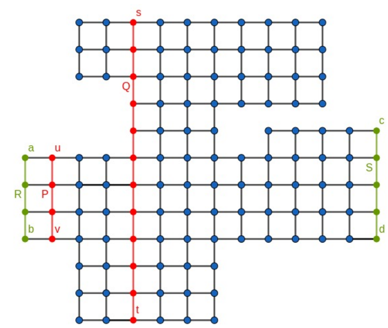


Figure 2.5.1: A solid grid. Here *P, Q, R* and *S* are some of the column paths, of which R and S are boundary paths.

1. P is a shortest u, v-path.

2. If P is not a boundary column path, then $G - P$ has at least two connected components.

We have the following lemma.

Lemma 2.5.1. *Let P be a column path in solid grid G and let S be one of the components of $G - P$. Then S has a unique column path P' adjacent to P.*

Proof. We will prove this by contradiction. Let P_1 and P_2 be two paths of component S that are adjacent to P, such that bottom most vertex of P_1 is in a higher row that top most vertex of P_2. Let the bottom most vertex of P_1 be u and top most vertex of P_2 be v. Also let the neighbours of u and v in P be u' and v' respectively. Note that (u, v) can not be an edge, by definition of column paths.

Let $v', x_1, \ldots, x_k, u'$ be the path between u' and v' in P and $u, y_1, \ldots, y_j, v$ be the shortest path between u and v in S. Then $u, y_1, \ldots, y_j$, v, v', x_1, $\ldots, x_k, u', u$ is an internal face of the solid grid and has area more than 1 (since (u, v) is not an edge). Since it is not possible in a solid grid, this leads to a contradiction.

Hence S can have only one column path P' that is adjacent to P and this proves our claim. □

From Lemma 2.5.1, we have the following observation, which is central to our strategy to capture $\mathcal{R}$ using two cops.

Observation 2.5.1. *Let P be a column path in solid grid G and let $\mathcal{R}$ be in one of the components of $G - P$, say S. Then if a cop is guarding the column path P' of S, that is adjacent to P, then $\mathcal{R}$ cannot leave the component S without being captured.*

Now we prove the following theorem.

Theorem 2.5.1. *The cop number of solid grids is two.*

Proof. We give a cop strategy to capture $\mathcal{R}$ using two cops. In this strategy, cops will reduce the robber territory after every finite number of steps, subsequently capturing the robber. Let the two cops be $\mathcal{C}_1$ and $\mathcal{C}_2$. Cops follow the following strategy.

1. $\mathcal{C}_1$ begins by guarding a column path P. If P is a boundary path, then $\mathcal{R}$ is restricted to $G - P$, else $\mathcal{R}$ is restricted to one of the connected components of $G - P$, say S.

2. Now, cops find the column path P' in S that is adjacent to P, and $\mathcal{C}_2$ guards P'. This restricts $\mathcal{R}$ to S and hence we can free $\mathcal{C}_1$, which was guarding P earlier.

   $\mathcal{R}$ is now restricted to S and a column path P' is guarded by $\mathcal{C}_2$. This further restricts $\mathcal{R}$ to either $S - P'$ (if P' is a boundary column path of S) or to one of the connected components of $S - P'$. Let the connected component $\mathcal{R}$ is restricted to be S'.

   This situation is same as situation in the end of step 1. So, we rename $\mathcal{C}_2$ as $\mathcal{C}_1$, $\mathcal{C}_1$ as $\mathcal{C}_2$, P' as P and S' as S, and repeat step 2.

Here cops reduce the robber territory in each step. Subsequently, the robber will be restricted to a single column and then $\mathcal{C}_2$ will capture $\mathcal{R}$.

To see the two cops are necessary to capture a robber in some solid grids, we can see that a cycle of 4 vertices, which is a solid grid, has cop number 2. □

2.5.2 Classical Cops and Robber on Partial Grids

In this section, we consider the game of classical Cops and Robber on the partial grids and prove that the cop number for the class of partial grids is 3. We have the following theorem.

Theorem 2.5.2. *Let $\mathcal{P}$ be the class of partial grids. Then $c(\mathcal{P}) = 3$.*

Proof. Aigner and Fromme [3] proved that the cop number for the class of planar graphs is 3. Since all partial grids are planar, $c(\mathcal{P}) \leq 3$. Hence to prove $c(\mathcal{P}) = 3$, it is sufficient to construct a partial grid graph G such that $c(G) = 3$.

Joret et al. [69] proved that for every positive integer r, subdividing each edge of a graph r times does not decrease the cop number. Thus, it would suffice to show that there exists some graph G' such that $c(G') = 3$ and after subdividing each edge of G' r times, we can get a partial grid graph G.

It is well known that the cop number of the dodecahedron graph is 3. It can be proved by the results of Aigner and Fromme [3] since it is a planar graph, and also a 3-regular graph with girth 5. We will show that by subdividing each edge of the dodecahedron an equal number of times, we can get a graph G which is a partial grid.

In Figure 2.5.2, we present a dodecahedron where vertices are marked from $1, \ldots, 20$. We obtain a graph G by subdividing each edge of the dodecahedron 19 times. In Figure 2.5.3, we give a partial grid representation of G. Each vertex i of the dodecahedron is a vertex i in G (and

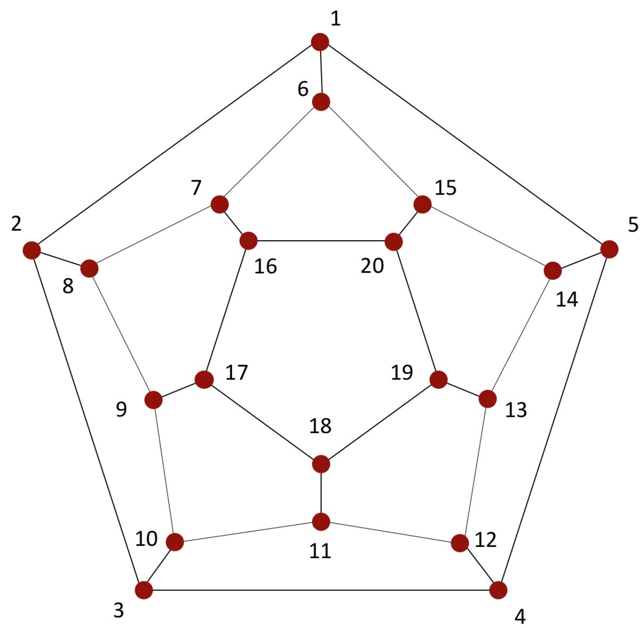


Figure 2.5.2: A dodecahedron labelling.

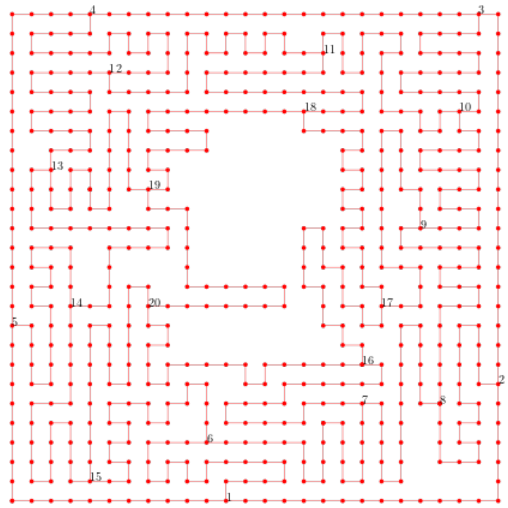


Figure 2.5.3: A partial grid obtained on subdividing each edge of dodecahedron 19 times.

there are new vertices too).

This completes the proof of Theorem 2.5.2. □

2.6 Concluding remarks and open problems

In this chapter, we study two variants of the cops and robber game, namely *Cops and attacking Robber* and *lazy Cops and Robber*. We find the attacking cop number of planar, cylindrical, and toroidal grids. We also bound the cop number for hypercubes and high-dimensional grids. We also find the lazy cop number of planar and cylindrical grids.

The classical version of the game is well studied on graph products. We study the game of Cops and attacking Robber on the Cartesian products of paths and cycles. It will be interesting to study cops and attacking robber game on general graph products as well as on the Cartesian product of trees. Moreover, the lazy cop number of toroidal grids remains open.

Question 2.6.1. *What is the lazy cop number for the finite toroidal grids?*

3

Cops, robbers and oriented graphs

Contents

In this chapter, we study the game of Cops and Robber on oriented graphs. We begin by giving a brief survey and recollecting the definitions relevant to this chapter in Section 3.1. We also compare the three graph parameters $c_n(\cdot), c_s(\cdot)$, $c_w(\cdot)$ (defined later in Section 3.1). In Section 3.2, we give an overview of the results discussed in this chapter. The normal, strong and weak cop models are studied in Sections 3.3, 3.4 and 3.5, respectively.

3.1 Preliminaries

3.1.1 Brief Survey

Here we present a brief survey of the game of Cops and Robber played on directed graphs. In most of these works, the rules of the game are similar to classical Cops and robber with the only difference that, in its turn, a player can move to an out-neighbour of the vertex occupied by the player. In this thesis, we refer to this model as *normal cop model*, and is defined formally later in this chapter.

Hamidoune [62] considered the Cops and Robber game on Cayley digraphs. Frieze et al. [51] studied the game on digraphs and gave an upper bound of $O\left(\frac{n(\log\log n)^2}{\log n}\right)$ for cop number in digraphs. Along these lines, Loh and Oh [77] constructively proved the existence of a strongly connected planar digraph with cop number greater than three. They also prove that every n-vertex strongly connected planar digraph has cop number at most $O(\sqrt{n})$.

Goldstein and Reingold [58] proved that deciding whether k cops can capture a robber is EXPTIME-complete if k is not fixed, and either the initial positions are given or the graph is directed. Later, Kinnersley [72] proved that determining the cop number of a graph or digraph is EXPTIME-complete. Kinnersley [73] also showed that n-vertex strongly connected cop-win digraphs can have capture time $\Omega(n^2)$.

Hahn and MacGillivray [61] gave an algorithmic characterization of the cop-win finite digraphs. They also showed that any k-cop game can be reduced to 1-cop game resulting in an algorithmic characterization for

k-cop-win finite digraphs. However, these results do not give a structural characterization of such graphs. Later Darlington et al. [40] tried to structurally characterize cop-win oriented graphs and gave a conjecture that was later disproved by Khatri et al. [71], who also studied the game in oriented outerplanar graphs and line digraphs.

Recently, the cop number of planar Eulerian digraphs and related families was studied in several articles [41, 63, 64]. In particular, Hosseini and Mohar [64] considered the orientations of integer grid that are vertex-transitive and showed that at most four cops can capture the robber on arbitrary finite quotients of these directed grids. De la Maza et al. [41] considered the *straight-ahead* orientations of 4-regular quadrangulations of the torus and the Klein bottle and proved that their cop number is bounded by a constant. They also showed that the cop number of every k-regularly oriented toroidal grid is at most 13. Furthermore, Bonato and Mohar [23] explored some future directions of research.

Bradshaw et al. [29] proved that the cop number of directed and undirected Cayley graphs on abelian groups has an upper bound of the form of $O(\sqrt{n})$. Modifying this construction they obtained families of graphs and digraphs with cop number $\Theta(\sqrt{n})$. The family of digraphs thus obtained has the largest cop number in terms of n of any known digraph construction.

Recently, in the open problem session of GRASTA 2014[1] [47], Nicolas Nisse introduced one of the variants (the strong cop model) in directed graphs (digraphs) and asked to characterize the "cop-win" graphs in two variants (the normal cop model and the strong cop model).

3.1.2 Preliminaries

An oriented graph is a directed graph without loops or directed 2-cycles. We start with an oriented graph $\overrightarrow{G}$ and Player 1 places k cops on its

[1]GRASTA 2014: http://www-sop.inria.fr/coati/events/grasta2014/

vertices. Multiple cops can occupy the same vertex v. After that Player 2 places the robber on one vertex of the graph. After the setup, Player 1 and 2 take turns to move the cops and robber, respectively, with Player 1 taking the first turn. Player 1 wins if after finitely many turns the robber and a cop are on the same vertex. In this case, we say that the cop *captures* the robber. Player 2 wins if Player 1 does not win in a finite number of moves.

Since $\overrightarrow{G}$ is an oriented graph, one can define two types of moves. In a *normal move* the cop or the robber can move along the arc, whereas in a *strong move* the cop or the robber can also move against the arc.

In the *normal cop model*, in each round, Player 1 can perform at most one normal move on each of its cops, whereas Player 2 can perform at most one normal move on the robber. In the *strong cop model*, in each round, Player 1 can perform at most one strong move on each of its cops, whereas Player 2 can perform at most one normal move on the robber. In the *weak cop model*, in each round, Player 1 can perform at most one normal move on each of its cops, whereas Player 2 can perform at most one strong move on the robber.

Next, we define a few necessary parameters. The *normal (respectively, strong, weak) cop number* $c_n(\overrightarrow{G})$ (respectively, $c_s(\overrightarrow{G})$, $c_w(\overrightarrow{G})$) of an oriented graph $\overrightarrow{G}$ is the minimum number of cops needed by Player 1 to have a winning strategy in the normal (respectively, strong, weak) cop model. Furthermore, for a family $\mathcal{F}$ of oriented graphs

$$c_x(\mathcal{F}) = \max\{c_x(\overrightarrow{G}) \mid \overrightarrow{G} \in \mathcal{F}\}$$

where $x \in \{n, s, w\}$. Given a fixed model, an oriented graph is *cop-win* if Player 1 has a winning strategy playing with a single cop.

Let uv be an arc of an oriented graph $\overrightarrow{G}$. We say that u is an *in-neighbor* of v and v is an *out-neighbor* of u. Let $N^-(u)$ and $N^+(u)$ denote the set of in-neighbors and out-neighbors of u, respectively. A vertex without any in-neighbor is a *source* and a vertex without any out-neighbor is a *sink*.

The *out-degree* of v is $d^+(v) = |N^+(v)|$ and its *in-degree* is $d^-(v) = |N^-(v)|$. Let $N^+[v] = N^+(v) \cup \{v\}$ denote the *closed out-neighbourhood* of v.

If a cop moves to an in-neighbor of the robber $\mathcal{R}$, then we say that the cop *attacks* the robber. The robber is on a *safe* vertex from a cop if it cannot be captured by that cop in the next turn of Player 1. The robber *evades* capture if every time the cop attacks it, $\mathcal{R}$ can move to a safe vertex.

We end this section with some basic results about the relation between strong, normal, and weak cop numbers of oriented graphs.

The first result follows directly from the definitions.

Proposition 3.1.1. *For any oriented graph $\overrightarrow{G}$ we have $c_s(\overrightarrow{G}) \leq c_n(\overrightarrow{G}) \leq c_w(\overrightarrow{G})$.*

Observe that there are plenty of oriented graphs, the transitive tournament for instance, where equality holds in each of the cases. However, it is interesting to study the gap between these parameters.

Proposition 3.1.2. *Given any $m, n \in \mathbb{N}$, there exists an oriented graph $\overrightarrow{G}$ such that $c_n(\overrightarrow{G}) - c_s(\overrightarrow{G}) = n$ and $c_w(\overrightarrow{G}) - c_n(\overrightarrow{G}) \geq m$.*

Proof. The oriented graph $\overrightarrow{G} = \overrightarrow{G}_{m,n}$ is composed of two oriented graphs $\overrightarrow{A}_n$ and $\overrightarrow{B}_m$. The oriented graph $\overrightarrow{A}_n$ is an orientation of the star graph such that its central vertex v is a sink having $n+1$ in-neighbors $v_0, \dots v_n$.

We know that there exist graphs with arbitrarily high cop number in the undirected case [3]. Let B_m be a connected undirected graph with cop number at least $m+1$. Let $\overrightarrow{B}_m$ be an orientation of B_m, such that it is a directed acyclic graph having a single source u. Such an orientation can be obtained by performing a breadth-first search (BFS) rooted at u and orienting each edge from the lower indexed vertex to

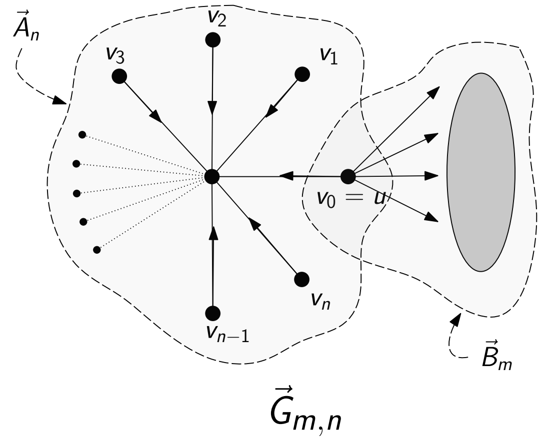


Figure 3.1.1: Illustration of $\overrightarrow{G}_{m,n}$.

the higher indexed vertex with respect to the BFS ordering. Note that $c_s(\overrightarrow{B}_m) = c_n(\overrightarrow{B}_m) = 1$ as the cop can start from u and keep following the robber who can move only a finite number of times (in the directed acyclic graph).

The graph $\overrightarrow{G}_{m,n}$ is obtained by *merging* vertices u and v_0 to form a single vertex v_{merge}. Observe that v_{merge} is a source in $\overrightarrow{G}_{m,n}$. In particular, after merging, the out-neighbors of both u and v_0 become out-neighbors of v_{merge}. See Figure 3.1.1 for reference.

Note that $c_s(\overrightarrow{G}_{m,n}) = 1$ as Player 1 can place one cop on v and capture the robber in one move if it is in $\overrightarrow{A}_n$ or capture the robber in a finite number of moves if it is in $\overrightarrow{B}_m$ as observed before.

Next let us prove that $c_n(\overrightarrow{G}_{m,n}) = n+1$. Indeed, we have $c_n(\overrightarrow{G}_{m,n}) \geq n+1$, as Player 1 must keep a cop on each of the $(n+1)$ sources $v_{merge}, v_1, \ldots, v_n$. Since $\overrightarrow{B}_m$ is a directed acyclic graph, the cop at v_{merge} can capture the robber if it is placed in $\overrightarrow{B}_m$.

Now we will show that $c_w(\overrightarrow{G}_{m,n}) \geq m+n+1$. Observe that Player 1 needs to place $(n+1)$ cops on each of the $(n+1)$ sources. This will ensure that if the robber is placed on any vertex of $\overrightarrow{A_n}$, then it will be captured.

However, if the robber is placed on a vertex of $\overrightarrow{B}_m$, then the only cop in $\overrightarrow{A_n}$ that may enter $\overrightarrow{B}_m$ is the cop placed on v_{merge}, as this is the weak cop model. Hence, we need to place at least m more cops on the vertices of $\overrightarrow{B}_m$ as the cop number of the undirected graph B_m is $(m+1)$. $\square$

3.2 Chapter overview

In Section 3.3, we study the normal cop number of oriented graphs. We begin by proving a Mycielski-type result by constructing oriented graphs with high normal cop number and girth. Then we attempt to characterize the cop-win oriented graphs in various graph families. It is easy to see that for an oriented graph to be cop-win, it must have a unique source, otherwise $\mathcal{R}$ can start at a source vertex not occupied by the cop and the cop then can never reach this vertex. Therefore, all graphs that we consider for being cop-win are assumed to have a unique source vertex. In particular, we show that an oriented triangle-free graph is cop-win if and only if it is a directed acyclic graph (DAG). As a corollary, it proves that oriented bipartite graphs are cop-win if and only they are DAG. We also prove a similar result for outerplanar graphs, proving that an oriented outerplanar graph is cop-win if and only if it is a DAG. For subcubic graphs other than K_4, we show that an oriented subcubic graph (other than K_4) is cop-win if and only if they are DAG.

In Section 3.4, we study the strong cop model on oriented graphs. We begin by proving that there exist graphs with arbitrarily high strong cop number. We also extend this result to bipartite graphs. Next, we consider the strong cop number of oriented planar graphs, outerplanar graphs, and series-parallel graphs. In particular, we prove that the strong cop number of outerplanar graphs is two. We also prove that a specific class of oriented outerplanar graphs whose weak dual is a collection of paths are strong cop-win. We also consider the strong cop model on oriented Cartesian grids and show that they are also strong cop-win.

In Section 3.5, we consider the weak cop model on oriented graphs and characterize the weak cop-win oriented graphs. For this, we use a technique similar to the cop-win characterization by Nowakowski and Winkler [86].

Finally, we draw conclusions in Section 3.6.

3.3 Normal Cop Model

It is known [51] that, in the normal cop model, if we can compute the cop number for strongly connected digraphs, then we can compute the cop number for weakly connected digraphs. Taking a cue from the above, we start by constructing strongly connected oriented graphs with arbitrarily high normal cop number, minimum degree, and *girth* (length of a smallest cycle of the underlying graph).

Theorem 3.3.1. *Given any $g \geq 5$ and $c \geq 3$, there exists a regular strongly connected oriented graph $\overrightarrow{G}_{g,c}$ with girth at least g and outdegree c having $c_n(\overrightarrow{G}_{g,c}) \geq c+1$.*

Proof. For this proof, we borrow a construction to form *regular expander graphs* with high girth from an unpublished note of J. Kilbane on graphs of large girth (currently this note is not available on the internet). We present their complete construction which uses the concept of *L-lifts* introduced by Amit and Linial [5].

Given a set L and a simple graph G, let G^L be a graph on the set of vertices $V(G^L) = V(G) \times L$. Moreover, let the edges of G^L satisfy the following property: for each edge $uv \in E(G)$, there is a perfect matching between the vertex subsets $\{u\} \times L$ and $\{v\} \times L$. Furthermore, other than the above-mentioned perfect matchings, there are no other edges in G^L. Such a G^L is an L-lift of G. Observe that there can be different L-lifts of a graph.

Given a path $u_1u_2\cdots u_k$ of G, its L-lift G^L will have many corresponding paths of the form (u_1, l_1) $(u_2, l_2)\cdots(u_k, l_k)$, where $l_1, l_2, \ldots, l_k \in L$. The path $u_1u_2\cdots u_k$ is called the *projection* of $(u_1, l_1)(u_2, l_2)\cdots(u_k, l_k)$.

Further, consider a graph G with m edges $e_1, e_2, \ldots, e_m$. Let $L = \{0, 1\}^m$, that is, the set of all binary vectors of length m. Let us now construct a graph G^L and then show that it is an L-lift of G. The vertex set of G^L is given by $V(G^L) = V(G) \times L$. Let $uv = e_i$ be an edge of G. Then, whenever we have two binary vectors $l, l' \in L$ which only differs at the i^{th} coordinate, we add the edges $(u, l)(v, l')$ and $(u, l')(v, l)$ in G^L. Observe that this ensures a perfect matching between the sets $\{u\} \times L$ and $\{v\} \times L$. Moreover, these are the only edges of G^L. Hence, G^L is indeed an L-lift of G.

Next, let us pick a shortest cycle C_0 in G^L. Its projection in G is also a cycle C. We claim that for every edge $uv \in E(C)$, there are at least two edges in C_0 between $\{u\} \times L$ and $\{v\} \times L$. To prove this claim, note that if we start at a vertex (u, l) of C_0, then the next vertex, say, (v, l'), is such that the two binary vectors l and l' differ only at their i^{th} coordinates. Now to reach (u, l) again, for completing the cycle, we need to flip the i^{th} coordinate of the binary vector once more. This can only happen if we traverse uv once again. Thus, $|C_0| \geq 2|C|$. Hence the girth of G^L is at least twice the girth of G.

To construct the oriented graph with arbitrarily high cop number and girth, we take a K_{2c+1} and go on applying the above-mentioned L-lift construction repeatedly until the girth is at least g. Denote the so-obtained graph as $G_{g,c}$. As L-lifts of a k-regular graph is also k-regular, the graph $G_{g,c}$ is $2c$-regular; and hence Eulerian. Now obtain the oriented graph $\overrightarrow{G}_{g,c}$ by making an Eulerian circuit of $G_{g,c}$ a directed circuit by assigning orientations to its edges. This results in a strongly connected oriented graph with girth at least g and the out-degree of each vertex is c. Thus its normal cop number is at least $(c + 1)$ as we know that a strongly connected oriented graph with girth at least five has normal cop number $c_n(\overrightarrow{G}) \geq \delta^+(\overrightarrow{G}) + 1$, where $\delta^+(\overrightarrow{G})$ is the minimum out-degree of $\overrightarrow{G}$ [77]. $\square$

Darlington et al. [40] characterized cop-win oriented paths and trees in the normal cop model. We are also going to do so for some other families of oriented graphs.

A *transitive-triangle-free* oriented graph is an oriented graph with no transitive triangles. The following theorem characterizes cop-win transitive-triangle-free oriented graphs, a superclass of triangle-free oriented graphs.

Proposition 3.3.1. *Let $\overrightarrow{G}$ be transitive-triangle-free oriented graph. Then $\overrightarrow{G}$ is cop-win if and only if $\overrightarrow{G}$ is a directed acyclic graph with one source.*

Proof. Observe that any directed acyclic graph with one source is cop-win, and every cop-win oriented graph has exactly one source. Thus it suffices to prove that if a transitive triangle-free oriented graph $\overrightarrow{G}$ is cop-win, then it is a directed acyclic graph.

Suppose $\overrightarrow{G}$ has a directed cycle $\overrightarrow{C}$ on at least 3 vertices. We will now give a strategy for the robber $\mathcal{R}$ to escape. Note that the cop $\mathcal{C}$ must be placed at the source initially, as otherwise Player 2 places $\mathcal{R}$ on the source and wins. The robber $\mathcal{R}$ initially places himself at some safe vertex of $\overrightarrow{C}$. Such a vertex exists, as any vertex in $\overrightarrow{G}$ cannot dominate two consecutive vertices in $\overrightarrow{C}$, else a transitive triangle is created. Now $\mathcal{R}$ moves to the next vertex in $\overrightarrow{C}$ whenever $\mathcal{R}$ lies in the out-neighbour of $\mathcal{C}$. Whenever $\mathcal{C}$ attacks $\mathcal{R}$, the robber moves to the next vertex in $\overrightarrow{C}$ and evades the attack. Since $\overrightarrow{C}$ is a directed cycle, $\mathcal{C}$ cannot capture $\mathcal{R}$. This contradicts that $\overrightarrow{G}$ is a cop-win graph, hence the result. $\square$

As bipartite graphs are triangle-free, we have the following corollary.

Corollary 1. *Let $\overrightarrow{G}$ be an oriented bipartite graph. Then $\overrightarrow{G}$ is cop-win if and only if $\overrightarrow{G}$ is a directed acyclic graph with one source.*

Next, we characterize the cop-win oriented outerplanar graphs.

Proposition 3.3.2. *Let $\overrightarrow{G}$ be an oriented outerplanar graph. Then $\overrightarrow{G}$ is cop-win if and only if $\overrightarrow{G}$ is a directed acyclic graph with one source.*

Proof. The 'if' part is obvious.

For proving the 'only if' part, first note that a graph cannot be cop-win if it has no source or at least two sources. Thus suppose that there exists an oriented outerplanar cop-win graph $\overrightarrow{G}$ containing a directed cycle $\overrightarrow{C}$ with exactly one source v. The cop c must be initially placed on the source v.

Note that at most two vertices of $\overrightarrow{C}$ can have a path made up of vertices from outside $\overrightarrow{C}$ connecting v in order to avoid a K_4-minor. Thus there is at least one safe vertex u in $\overrightarrow{C}$ such that any directed path connecting v to u must go through some vertex of $\overrightarrow{C}$ other than u. Hence if the robber $\mathcal{R}$ places itself on u and does not move until c comes on a vertex of $\overrightarrow{C}$, it cannot be captured.

If c is on a vertex of $\overrightarrow{C}$ and starts moving towards $\mathcal{R}$ following the direction of the arcs of $\overrightarrow{C}$, then $\mathcal{R}$ also moves forward and evades c.

Thus c must go out of $\overrightarrow{C}$ in order to try and capture $\mathcal{R}$. The moment c goes out to some vertex w outside $\overrightarrow{C}$, then $\mathcal{R}$ is either on a safe vertex or it can move to a safe vertex on $\overrightarrow{C}$ in its next move as w can be adjacent to at most two vertices of $\overrightarrow{C}$ in order to avoid a K_4-minor.

This brings us to a situation similar to the initial situation. Thus the robber will always evade the cop, a contradiction. □

Proposition 3.3.3. *Let $\overrightarrow{G}$ be an oriented subcubic[2] graph other than K_4. Then $\overrightarrow{G}$ is cop-win if and only if $\overrightarrow{G}$ is a directed acyclic graph with one source.*

Proof. The 'if' part is obvious.

For proving the 'only if' part, first note that a graph cannot be cop-win if it has no source or at least two sources. Thus suppose that there

[2]An oriented graph whose underlying undirected graph is a subcubic graph.

exists an oriented subcubic cop-win graph $\overrightarrow{G}$, other than K_4, containing a directed cycle $\overrightarrow{C}$ with exactly one source v. The cop c must be initially placed on the source v.

Consider a shortest directed cycle $\overrightarrow{C} = v_0v_1 \dots v_kv_0$ of $\overrightarrow{G}$. Since $\overrightarrow{G}$ is a subcubic graph and $\overrightarrow{C}$ is a smallest directed cycle; each $v_i \in \overrightarrow{C}$ can have at most one more neighbor not belonging to $V(\overrightarrow{C})$. Also since the graph is subcubic and v is the unique source, any vertex $u \neq v$ can have at most two out-neighbors in $\overrightarrow{C}$.

On the other hand, the source vertex v can have at most three out-neighbors in $\overrightarrow{C}$. As we know that the underlying graph of our graph is not K_4, there exists a vertex of $\overrightarrow{C}$ which is not an out-neighbor of v.

Consider the possibility that there exists a vertex $u \notin V(\overrightarrow{C})$ (and $u \neq v$) containing both v_{i-1} and v_i as its out-neighbors for some $i \in \{0, 1, \cdots, k\}$. Here the $+, -$ operations on the indices of the vertices of $\overrightarrow{C}$ are considered modulo $(k+1)$. If a vertex such as u does not exist, then $\mathcal{R}$ will evade c by moving along the cycle $\overrightarrow{C}$, whenever under attack.

Therefore, such a u, which has both v_{i-1} and v_i as its out-neighbors, exists. Here we call v_i a *special vertex*. There may be more than one special vertex in $\overrightarrow{C}$. The strategy of $\mathcal{R}$ is to initially start on a special vertex of $\overrightarrow{C}$ and move towards the next special vertex of $\overrightarrow{C}$ when under attack. We will now show that this strategy works.

The robber $\mathcal{R}$ starts at a special vertex v_i and does not move until c attacks (either from v_{i-1} or u). When c attacks, $\mathcal{R}$ moves to v_{i+1}. At this point $\mathcal{R}$ is two vertices ahead of c, and $\mathcal{R}$ keeps moving towards the next special vertex in $\overrightarrow{C}$, irrespective of c's moves. If there are no more special vertices, then $\mathcal{R}$ traverses $\overrightarrow{C}$ completely and reaches v_i (the vertex v_i itself is the next special vertex, in this case). In any case $\mathcal{R}$ maintains a distance of at least two from c (just before c's move), since $\overrightarrow{C}$ is a shortest directed cycle. Thus $\mathcal{R}$ can forever evade capture. This contradicts our initial assumption that $\overrightarrow{G}$ is a cop-win graph. $\square$

3.4 Strong Cop model

The strong cop number of an oriented graph is upper bounded by the cop number in the classical version of the game on the underlying undirected graph. The following observation is trivially true.

Observation 3.4.1. *For any oriented graph $\overrightarrow{G}$, we have $c_s(\overrightarrow{G}) \leq c(G)$.*

Given a simple graph G, we are going to describe a specific construction of an oriented graph $\overrightarrow{S_G}$ and study the relation between $c(G)$ and $c_s(\overrightarrow{S_G})$. Later we proceed to find the strong cop number of oriented planar graphs, outerplanar graphs, and series-parallel graphs.

Construction: Given an undirected graph G, first replace each of its edges with a directed 2-cycle. After that subdivide each arc to obtain an oriented graph. That is, each edge v_iv_j of G is replaced by a directed 4-cycle $v_iu_{ij}v_ju_{ji}v_i$ to obtain the oriented graph $\overrightarrow{S_G}$.

We have the following lemma that relates the strong cop number of $\overrightarrow{S_G}$ with the cop number of G.

Lemma 3.4.1. *For any simple graph G, $c_s(\overrightarrow{S_G}) \geq c(G)$.*

Proof. We know that $c_s(\overrightarrow{S_G})$ cops have a strategy to capture the robber $\mathcal{R}$ in $\overrightarrow{S_G}$. We will show that $c_s(\overrightarrow{S_G})$ cops have a winning strategy in G as well. To be more specific, we will use the winning strategy of $c_s(\overrightarrow{S_G})$ cops in $\overrightarrow{S_G}$ to obtain a winning strategy in G.

As the game is played in G, we also play it simultaneously in $\overrightarrow{S_G}$. The moves of the cops in $\overrightarrow{S_G}$, following the winning strategy, are translated to a winning strategy in G. We will describe the procedure to do so below. Before that, we will present the notion of image.

Note that the sets $N^+[v]$, for $v \in V(G)$, partition $V(\overrightarrow{S_G})$. That means, for any $x \in V(\overrightarrow{S_G})$, there exists a unique $v \in V(G)$ such that $x \in N^+[v]$ in $\overrightarrow{S_G}$. Now define the image of x as $i(x) = v$.

Initially place $c_s(\overrightarrow{S_G})$ cops in $\overrightarrow{S_G}$ according to the winning strategy. Correspondingly place $c_s(\overrightarrow{S_G})$ cops in G on the images of the vertices of $\overrightarrow{S_G}$ where we had placed the cops. Secondly, place $\mathcal{R}$ in G. In $\overrightarrow{S_G}$ also place $\mathcal{R}$ on the exact same vertex.

For each move of the robber in G, make two moves of the robber in $\overrightarrow{S_G}$. To be precise, if the robber moves from v_i to v_j in G, then the robber will move from v_i to u_{ij} and then to v_j in $\overrightarrow{S_G}$.

The cops will move according to the winning strategy in $\overrightarrow{S_G}$. Correspondingly, we will move the cops in G. To be precise, if a cop moves from x to y in $\overrightarrow{S_G}$ in two consecutive turns, chasing the robber, then correspondingly we move the cop from $i(x)$ to $i(y)$ in G. This is possible as $i(x)$ and $i(y)$ are either adjacent in G or the same vertex in G.

Notice that, when $\mathcal{R}$ is captured in $\overrightarrow{S_G}$, it is also captured in G. □

Remark 3.4.1. *Berarducci and Intrigila [12], and Joret, Kaminski, and Theis [69] showed that subdividing every edge of an undirected graph a fixed number of times does not decrease the cop number. Lemma 3.4.1 is a weak analogue of their result. Our proof is similar to that in [12] and can be generalized to prove that subdividing every arc of an oriented graph a fixed number of times does not decrease its strong cop number.*

As a result of Lemma 3.4.1, we find the strong cop number of oriented planar graphs and then form oriented graphs with arbitrarily high strong cop number.

Corollary 2. *The strong cop number of the family of oriented planar graphs is three.*

Proof. We know that the cop number of planar graphs is three [3]. Thus, by Observation 3.4.1, the strong cop number of oriented planar graphs is at most 3. On the other hand, if H is a planar graph with cop number 3 (for example, dodecahedron), then, by Lemma 3.4.1, the strong cop number of $\overrightarrow{S_H}$ is at least 3. Thus we are done by observing that $\overrightarrow{S_H}$ is planar. □

It is known that the cop number of bipartite graphs is unbounded [24]. We have a similar result for the strong cop model also.

Corollary 3. *For every $k \in \mathbb{N}$, there exists an oriented bipartite graph $\overrightarrow{H_k}$ such that $c_s(\overrightarrow{H_k}) \geq k$.*

Proof. We know that there exists a graph H_n with cop number at least k [3]. Now take $\overrightarrow{H_n} = \overrightarrow{S_{H_n}}$. Note that $\overrightarrow{S_{H_n}}$ is bipartite. Thus, the proof follows from Lemma 3.4.1. $\square$

Next, we find the strong cop numbers of the family of oriented outerplanar and series-parallel graphs.

Theorem 3.4.1. *The strong cop number of the family of oriented outerplanar graphs is two.*

Proof. The cop number of outerplanar graphs in the classical game on undirected graphs is two [34]. Hence, by Observation 3.4.1, it suffices to construct an oriented graph that is not strong cop-win.

Construction: Consider the cycle $\overrightarrow{C}_1$, depicted in Fig. 3.4.1, on vertices $v_0, v_1, \cdots, v_{23}$ vertices having *cycle arcs* $v_i v_{i-1}$ and *chord arcs* $v_{2i}v_{2i+2}$. The $+$ and $-$ operations on the indices of $V(\overrightarrow{C}_1)$ are taken modulo 24. Consider another copy $\overrightarrow{C}_2$ of $\overrightarrow{C}_1$. In $\overrightarrow{C}_2$ rename each vertex v_i as u_i. Now merge the vertex v_0 of $\overrightarrow{C}_1$ with the vertex u_0 of $\overrightarrow{C}_2$ to obtain the graph $\overrightarrow{C}$.

We give a robber-win strategy for $\overrightarrow{C}$ when there is just one strong cop in this oriented graph.

Initial Setup If the cop c is placed at v_0, then robber $\mathcal{R}$ enters at v_4; else $\mathcal{R}$ starts at v_4 or u_4 depending on whether c starts at C_2 or C_1, respectively. In the latter case, $\mathcal{R}$ passes its moves until c is at v_0 (in order to catch $\mathcal{R}$, c has to go through v_0). Once c reaches v_0, $\mathcal{R}$ passes its

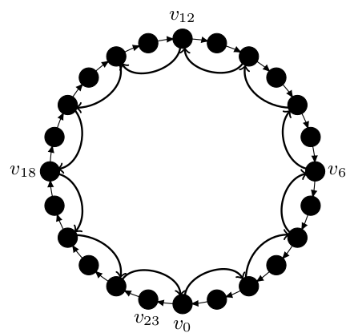


Figure 3.4.1: The biconnected outerplanar graph $\overrightarrow{C}_1$.

move once more; reducing this case to the former case. Hence, without loss of generality, assume that $\mathcal{C}$ and $\mathcal{R}$ start at v_0 and v_4, respectively.

In the rest of this proof, we show that if $\mathcal{C}$ tries to capture $\mathcal{R}$, then $\mathcal{R}$ reaches the *initial configuration* ($\mathcal{C}$ at v_0 and $\mathcal{R}$ at v_4) or its *equivalent configuration* ($\mathcal{C}$ at v_0 and $\mathcal{R}$ at u_4) without being captured. Precisely, we show that if $\mathcal{C}$ pursues $\mathcal{R}$, then $\mathcal{R}$ reaches v_0 two turns before $\mathcal{C}$ (or the game continues indefinitely); thereby evading capture indefinitely.

To simplify our presentation, we use the following notations. Read $X(U * V)$ as "X moves from U to V in $*$ sense". Read $X(*)$ as "X moves in $*$ sense to an adjacent vertex", where $* \in \{\circlearrowleft, \circlearrowright\}$, that is, counter-clockwise and clockwise, respectively. Let d_c denote the distance between $\mathcal{C}$ and $\mathcal{R}$ at the given instant in the underlying undirected graph.

Rules for the Robber: All operations are performed under modulo 24.

- *R0*: At any turn, if $\mathcal{C}$ passes its move then $\mathcal{R}$ passes its move; else $\mathcal{R}$ moves according to the following rules (*R1* – *R4*).
- *R1*: For $i = 1$ to 7, if $\mathcal{C}(v_{2i-4} \circlearrowleft v_{2i-2})$ or $\mathcal{C}(v_{2i-3} \circlearrowleft v_{2i-2})$ then $\mathcal{R}(v_{2i} \circlearrowleft v_{2i+2})$; else it passes its move.
- *R2*: For $i = 8$ to 11, $\mathcal{R}(v_{2i} \circlearrowleft v_{2i+2})$ irrespective of $\mathcal{C}$'s move.
- *R3*: If $\mathcal{R}$ is at v_{2i}, for $i \leq 7$, and $\mathcal{C}(\circlearrowright)$, then

- if d_c increases to at least 4, then $\mathcal{R}(\circlearrowright)$.
- if d_c increases but remains less than 4, then $\mathcal{R}$ passes its move.
- if d_c decreases, then $\mathcal{R}(\circlearrowright)$.

R4: If $\mathcal{R}$ is at v_{2i+1}, for $i < 7$, then $\mathcal{R}(\circlearrowright)$ irrespective of c's move.

The scenarios that are not addressed by the above rules do not occur.

Analysis We claim that $\mathcal{R}$ satisfies its objective, that is, $\mathcal{R}$ reaches v_0 at least two turns before c. Once $\mathcal{R}$ is at v_{14} and $c(v_{10} \circlearrowleft\ v_{12})$, then $\mathcal{R}$ keeps on moving counter-clockwise and reaches v_0 at least two turns before c (irrespective of c's moves). However if $c(\circlearrowright)$ and if d_c increases to at least 4, then $\mathcal{R}(\circlearrowright)$; else if $d_c < 4$, then $\mathcal{R}$ passes its move. The restriction $d_c \geq 4$ ensures that if c moves counter-clockwise, then $\mathcal{R}$ can safely move clockwise to the next vertex with an even index. For subsequent steps, if $c(\circlearrowright)$ and $\mathcal{R}$ is on v_{2i}, for $i = 1$ to 7, then $\mathcal{R}(\circlearrowright)$, provided the restrictions in *R3* are met. In any intermediate step if $c(\circlearrowleft)$, then $\mathcal{R}(\circlearrowright)$ if it is at a vertex with odd index; else $\mathcal{R}(\circlearrowleft)$ or $\mathcal{R}$ passes its move depending on whether c attacks it or not. In such a case $\mathcal{R}$ always stays at least two moves away from c and hence evades capture.

The only way left for c to capture $\mathcal{R}$ is if c continues moving counter-clockwise along the chord arcs and then tries to capture $\mathcal{R}$ which now moves counter-clockwise along the cycle arcs. However, in such a case also, one can check that $\mathcal{R}$ reaches v_0 at least two moves before c. Hence the constructed graph is not strong cop-win. □

Remark 3.4.2. *The analysis for the above non-strong cop-win graph (a graph that is not cop-win in the strong cop model) can be extended to a family of non-strong cop-win graphs built on the same idea. In the construction of the above graph take C_1 and C_2 as the cycles with an even length of at least* 24.

It is known that the cop number of series-parallel graphs in the classical game on undirected graphs is two [103]. Since the outerplanar graph in Figure 3.4.1 is also a series-parallel graph, by Observation 3.4.1, we have the following corollary.

Corollary 4. *The strong cop number of oriented series-parallel graphs is two.*

As mentioned earlier all the oriented graphs whose underlying graphs are cop-win graphs in the classical (undirected graph) version are strong cop-win. Next, we find some families of oriented graphs which are strong cop-win but whose underlying undirected graphs are not cop-win in the classical version. We begin with a specific class of outerplanar graphs. First, we need the following definition.

The *weak dual* of a plane graph G is a graph that has a vertex for each bounded face of G and two vertices are adjacent if the corresponding faces share an edge.

Theorem 3.4.2. *Oriented connected outerplanar graphs whose weak dual is a collection of paths are strong cop-win.*

Proof. Let $\overrightarrow{G}$ be an oriented outerplanar graph on n vertices such that the weak dual of its underlying graph G is a collection of paths. We call the edges in the outer face of G as *cycle edges* and the other edges as *chord edges*. Every edge in the weak dual of G corresponds to a chord edge in G, so there are two bounded faces incident to a chord edge, one on each side of it. Every bounded face in G has at most two chord edges, or else a claw[3] will be induced in the weak dual of G.

Let F be a bounded face in G, Also, let us assume that the robber $\mathcal{R}$ is on the vertex v (say) at the moment. Let the length of a shortest directed path from v to a vertex of F be l. The *image* $i_F(\mathcal{R})$ of the robber $\mathcal{R}$, is the set of vertices in F such that there is a directed path of length l

[3] A *claw* is a star graph on four vertices.

from v to them. The image $i_F(\mathcal{R})$ may possibly be empty if there are no directed paths from v to F. Note that, if v is not in F, then the vertices of $i_F(\mathcal{R})$ are incident to a chord edge. This also implies that $|i_F(\mathcal{R})| \leq 2$, as the weak dual of G is a collection of paths. Also if $|i_F(\mathcal{R})| = 2$, then the vertices from $i_F(\mathcal{R})$ are incident to the same chord edge.

If $|i_F(\mathcal{R})| = 2$, then after $\mathcal{R}$'s move, $i_F(\mathcal{R})$ may remain the same, or reduce to one vertex, or become empty. If $|i_F(\mathcal{R})| = 1$, then after $\mathcal{R}$'s move, $i_F(\mathcal{R})$ may remain the same, or become empty, or get changed to a set containing an adjacent vertex of x, where x is the vertex in $i_F(\mathcal{R})$ earlier. If $|i_F(\mathcal{R})| = 0$, then it remains the same irrespective of $\mathcal{R}$'s move.

As mentioned above, if $\mathcal{R}$ is not in F, then each vertex in $i_F(\mathcal{R})$ is an end vertex of a chord edge in F. Thus even if $\mathcal{R}$ moves, $i_F(\mathcal{R})$ might change to an adjacent vertex only (from one end vertex of a chord edge to the other end vertex if $\mathcal{R}$ is not in F or follows $\mathcal{R}$ if it is in F).

Now we give a strong cop-win strategy in $\overrightarrow{G}$. Select a face F_1 in $\overrightarrow{G}$ and place the cop c in some vertex of F_1. After $\mathcal{R}$ is placed in $\overrightarrow{G}$, we find $i_{F_1}(\mathcal{R})$. If $i_{F_1}(\mathcal{R}) = \emptyset$, or becomes $\emptyset$ at any point, then c moves to the other face adjacent to F_1 that is *closer* to $\mathcal{R}$.

If $i_{F_1}(\mathcal{R}) \neq \emptyset$, then capture a vertex in $i_{F_1}(\mathcal{R})$. This is always possible as a vertex in $i_{F_1}(\mathcal{R})$ changes at most to an adjacent vertex. Now c is at an end vertex of a chord edge, say pq. Let F_2 be the other face incident at the chord edge. The cop c follows the same strategy as it did in F_1, but with with the following exception (in order to forbid $\mathcal{R}$ from entering F_1).

If $i_{F_2}(\mathcal{R}) = \{x\}$ and there are two dipaths from x to p and q, then c tries to capture $i_{F_2}(\mathcal{R})$ by going against the orientation on the shortest dipath. But if $\mathcal{R}$ enters F_2 and takes the other directed path, then c reverses back to pq and then takes the other dipath to x. This way c captures $\mathcal{R}$.

If $i_{F_2}(\mathcal{R}) = \{x, y\}$ and there are two dipaths from say x to p and y to q, then c tries to capture one of $i_{F_2}(\mathcal{R})$ by going against the orientation on the shortest dipath. But if $\mathcal{R}$ enters F_2 and takes the other directed path, then c reverses back to pq and then takes the other dipath. This way c captures $\mathcal{R}$. Following this strategy, $\mathcal{R}$ is forbidden to enter the explored faces by the cop.

At each iteration, we go on removing one face of the graph to which $\mathcal{R}$ cannot enter. Eventually both $\mathcal{R}$ and c end up in the same cycle, where $\mathcal{R}$ gets captured. $\square$

Our next class of strong cop-win graphs are oriented grids.

Theorem 3.4.3. *Oriented grids are strong cop-win.*

Proof. Fix a $m \times n$ grid $\overrightarrow{G}$ with vertices at $\{(i, j) \mid 0 \leq i \leq m-1, 0 \leq j \leq n-1\}$ in the usual coordinate system. The cop c starts at $(0, 0)$. We say that $\mathcal{R}$ is *restricted* to T_i, if $\mathcal{R}$ cannot move to a vertex of rows from 0 to $i-1$ of $\overrightarrow{G}$. Let (i, j) be the position of $\mathcal{R}$, then we define *image* of $\mathcal{R}$ in row r as (i, r). When c is at image of $\mathcal{R}$ in row r, we say that c is *guarding* row r. Observe that if c is guarding row r and $\mathcal{R}$ is in a row $r' > r$, then $\mathcal{R}$ is restricted to T_{r+1}.

Cop c begins by guarding row 0. Next we show that if c is guarding row $r < n-2$ (and so $\mathcal{R}$ is restricted to T_{r+1}), then after a finite number of moves c can guard row $r+1$ restricting $\mathcal{R}$ to T_{r+2}. Let $\mathcal{R}$ be at a vertex (i, j) and c is guarding row $r < j$. Now, if $\mathcal{R}$ moves up or down, then c moves up and guards row $r+1$. If $\mathcal{R}$ moves left or right, then c also moves left or right, respectively, and retains the guard of row r. Since the grid is oriented and $\mathcal{R}$ can make only weak moves, if $\mathcal{R}$ moves from a vertex u to v, then it cannot move from v to u. Hence, $\mathcal{R}$ can make at most m left and right moves and then has to either make an up or down move or skip a move. In both cases, c moves up and guards row $r+1$. Also, note that until c guards row $r+1$, it is guarding row r and hence $\mathcal{R}$ is always restricted to T_{r+1} during this process.

This way c finally guards row $n-2$ with $\mathcal{R}$ restricted to T_{n-1} (that is, row $n-1$). Again, since $\mathcal{R}$ can only move a finite number of times in row

$n-1$ and also $\mathcal{R}$ cannot move to row $n-2$, there will be no more moves for $\mathcal{R}$ and c will capture $\mathcal{R}$. □

3.5 Weak Cop Model

A vertex u in a directed graph is said to be a *corner vertex*, if there exists a vertex v such that $N^+[u] \cup N^-(u) \subseteq N^+[v]$, where $N^*[v] = N^*(v) \cup \{v\}$ for each $* \in \{+, -\}$. We also say that v *dominates* u.

Now we characterize all cop-win directed graphs in this model, which is adapted from the cop-win characterization of undirected graphs [86].

Theorem 3.5.1. *A directed graph is cop-win in the weak cop model if and only if by successively removing corner vertices, it can be reduced to a single vertex.*

In order to prove Theorem 3.5.1, we need the following two lemmas.

Lemma 3.5.1. *If a directed graph has no corner vertex, then it is not weak cop-win.*

Proof. Let $\overrightarrow{G}$ have no corner vertex. The robber $\mathcal{R}$ starts from a vertex that is not an out-neighbour of the cop c. The robber does not move unless c attacks it. Whenever $\mathcal{R}$ is under attack, it can move to a vertex that in not an out-neighbour of c (as there are no corner vertices in $\overrightarrow{G}$). Hence $\mathcal{R}$ never gets caught. □

Lemma 3.5.2. *A directed graph $\overrightarrow{G}$ with a corner u is weak cop-win if and only if $\overrightarrow{H} = \overrightarrow{G} \setminus \{u\}$ is weak cop-win.*

Proof. Let vertex v dominate u in G. Suppose $\overrightarrow{H}$ is cop-win. Define the image $i_{\mathcal{R}}$ of the robber $\mathcal{R}$ as follows: $i_{\mathcal{R}}(u) = v$ and $i_{\mathcal{R}}(x) = x$ for all $x \in V(\overrightarrow{H})$. So $i_{\mathcal{R}}$ is restricted to $\overrightarrow{H}$ and it can be captured by the cop c. If $\mathcal{R}$ is not on u, then it is captured. If $\mathcal{R}$ is on u, then c is on v and will capture $\mathcal{R}$ in its next move.

Suppose, on the other hand, $\overrightarrow{H}$ is not weak cop-win. Define the image $i_{\mathcal{C}}$ of the cop $\mathcal{C}$ as follows: $i_{\mathcal{C}}(u) = v$ and $i_{\mathcal{C}}(x) = x$ for all $x \in V(\overrightarrow{H})$. So $i_{\mathcal{C}}$ is restricted to $\overrightarrow{H}$ and $\mathcal{R}$ has a winning strategy against $i_{\mathcal{C}}$. If $\mathcal{C}$ is not on u, then $\mathcal{R}$ follows its winning strategy and does not get captured in $\mathcal{C}$'s next move. If $\mathcal{C}$ is on u, then $\mathcal{R}$ follows its winning strategy assuming $\mathcal{C}$ is on $i_{\mathcal{C}}(u) = v$. Since $\mathcal{R}$ has a winning strategy against $\mathcal{C}$ if $\mathcal{C}$ were at v instead, $\mathcal{R}$ does not get captured in $\mathcal{C}$'s next move (as v dominates u). So $\mathcal{R}$ evades capture; hence $\overrightarrow{G}$ is not weak cop-win. □

Finally, we are ready to prove Theorem 3.5.1.

Proof of Theorem 3.5.1. Lemma 3.5.2 implies that upon removing the corner vertices, the weak cop-win property of the graph remains the same. Now remove all possible corner vertices successively in the directed graph. If we end up with a single vertex, then it is weak cop-win. Otherwise, we end up with some other graph that has no corner vertices, Lemma 3.5.1 implies that it is not weak cop-win. □

3.6 Conclusion

In this chapter, we focus on three variants of the Cops and Robber game on oriented graphs. We are able to characterize the weak cop-win graphs. For the normal model, we find some cop-win graphs. For the strong model, we also find some cop-win graphs; and then find the strong cop numbers of some families of oriented graphs, namely, the outerplanar, the series-parallel, and the planar.

In the strong cop model, it will be interesting to study graph classes that have cop number two in the classical undirected game, such as, interval filament graphs, circular-arc graphs, function graphs. Also, the characterization of strong cop-win graphs remains open. Another interesting question is to construct oriented graphs that can have arbitrarily high strong cop number, minimum degree, and girth.

4

Cops and Robber on Intersection graphs

Contents

In this chapter we study the game of Cops and Robber on some classes of intersection graphs. In particular, we study the cop number of string graphs and rectangle intersection graphs.

4.1 Chapter overview

In Section 4.2, we provide some basic definitions and preliminary results. Our main observation of guarding a special path is given in Section 4.3.1, which is used to prove the improved upper bound on the cop number of string graphs in Section 4.3. In Section 4.4, we improve the bounds on the cop number of boxicity 2 graphs. We end with Section 4.5 where we suggest some future directions.

4.2 Preliminaries

Recollected definitions: All graphs considered in this chapter are finite, connected, and simple. Let $G(V, E)$ be a graph with vertex set V and edge set E. Let u be a vertex of G. Then the *open neighbourhood* of u is denoted by $N(u)$, and $N(u) = \{v : uv \in E\}$. The *closed neighbourhood* of u is denoted by $N[u]$, and $N[u] = N(u) \cup u$. For a subgraph H of G, we represent the closed neighbourhood of H by $N[H]$, where $N[H] = \bigcup_{v \in V(H)} N[v]$. We also define $G \setminus H$ as the graph induced by the vertices in G but not in H. For a vertex v of G, by $G \setminus v$ we refer to the graph induced by vertices of $V \setminus v$. Let $V' \subseteq V$ be a set of vertices of G. Then $G|_{V'}$ denote the subgraph of G induced by vertices of V'.

Consider a graph G and a subgraph H of G. We say that the robber $\mathcal{R}$ is *restricted* to H, if $\mathcal{R}$ cannot leave the vertices of H without getting captured. Here H is the *robber territory*. We say that cops *guard* H if the robber cannot enter the vertices of H without getting captured, by one of the cops guarding H, in the next cop move.

For vertices u and v in G, a path from u to v is denoted as u, v-path. For two vertices u and v in G, let $d(u, v)$ denote the length of a shortest u, v-path

New definitions: Let P be a shortest u_0, u_k-path such that $P = u_0, u_1, \ldots, u_k$. For $0 \leq i \leq k$, let $D_i = \{x \mid d(u_0, x) = i\}$ if $i < k$, and $D_i = \{x \mid d(u_0, x) \geq i\}$ otherwise. Hence, $u_i \in D_i$, for $i \leq k$.

$G(V, E)$ is an *intersection graph* if each vertex $v \in V$ corresponds to a set $\psi(v)$, and $(u, v) \in E$ if and only if $\psi(u) \cap \psi(v) \neq \emptyset$. A string graph $G = (V, E)$ is an intersection graph of *strings*, where each string $\psi(v)$ is a continuous image of the interval $[0, 1]$ into $\mathbb{R}^2$. Given a string graph G, we can generate strings corresponding to each vertex of V such that two strings intersect if and only if the corresponding two vertices are adjacent in G. These strings are said to be a *realization/representation* of graph G. We assume that the strings are non self-intersecting.

Segments, Faces and Regions: A set $A \subset \mathbb{R}^2$ is *arc-connected* if for any two points $a, b \in A$, the set A contains a curve with endpoints a and b.

Consider a fixed string representation Ψ of G. If two strings π and π' intersect at a point p, then we call p as an *intersection point*. In a fixed representation of a string graph G, a string can have multiple intersection points and two strings can have multiple intersection points in common. A *segment* s of a string π is a maximal continuous part of the string π that does not contain any intersection point other than its endpoints. A string containing k intersection points has $k + 1$ segments.

A *region* is an arc-connected area bounded by some segments of a set of strings in a string representation. A region also includes its boundary. Whenever we mention about region, it should satisfy our region definition. A *face* is a region not containing any intersection point between two strings except on the boundary and no string has a continuous part in the region that intersects the boundary of the region more than once. It

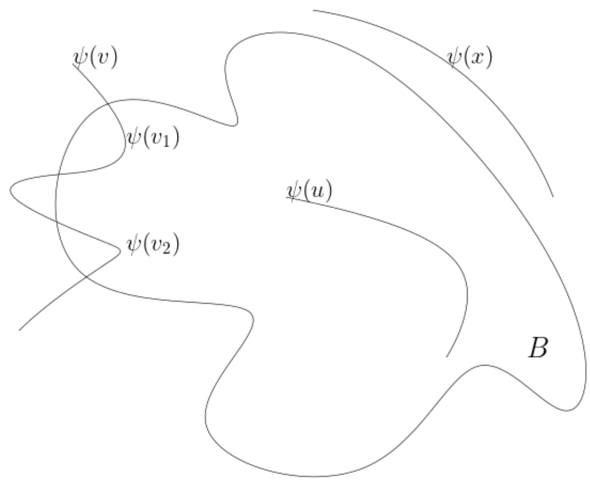


Figure 4.2.1: Here $\psi(u)$ is in Ψ_B, $\psi(x)$ is not in Ψ_B, and for $\psi(v)$, strings $\psi(v_1)$ and $\psi(v_2)$ are in Ψ_B.

is a standard assumption that for a finite string graph G, we can have a representation such that the number of segments and faces is finite.

Consider a region B of representation Ψ. We define the *representation restricted to* B, denoted by Ψ_B, in the following manner. If a string π is completely inside B, then we have π in Ψ_B also. If π is completely outside B, then π is not in Ψ_B. If a string π is such that some portion of π is outside B and some portion of π is inside B, then we do the following. Let $s_1, \ldots, s_k$ be the portions of the string π such that each endpoint of s_i, for $0 < i \leq k$, is either on the boundary of B or is an endpoint of the string π, and $s_i \in Phi_B$. Then instead of string π, we include k new strings. We consider each portion s_i, for $0 < i \leq k$, as a new string π_i in Ψ_B. See Figure 4.2.1 for illustration. Let $G_B(V_B, E_B)$ be the string graph corresponding to the representation Ψ_B. Here V_B is defined by the strings in Ψ_B and E_B is defined by the intersection between these strings. Observe that, though G_B might contain more vertices than G, the number of vertices in G_B remains finite. Moreover, the number of faces and the number of segments in Ψ_B are not more than that in Ψ.

Here we also say that G_B is the graph G *restricted* to region B.

Let Ψ be a fixed representation of a string graph G. Consider a curve C in the representation Ψ. C is composed of some of the segments of the strings from Ψ. Let C be composed of segments $s_1, \ldots, s_l$. Furthermore, consider a path P in G, such that $P = u_1, \ldots, u_k$. Suppose each segment $s \in \{s_1, \ldots, s_l\}$ is a segment of some string $\psi(u)$, $u \in \{u_1, u_2, \ldots, u_k\}$, and for each string $\psi(u)$, $u \in \{u_1, u_2, \ldots, u_k\}$, there is a segment $s \in \{s_1, \ldots, s_l\}$ such that s is a segment of $\psi(u)$. Here we say curve C is *related* to path P. Observe that $l \geq k$. Note that multiple curves may relate to the same path, and a curve may be related to multiple paths. For example, consider a complete graph K_n (which is a string graph) and a string representation of K_n, denoted by $\Psi(K_n)$. If we choose a curve such that it contains at least one segment from each string of $\Psi(K_n)$, then this curve corresponds to every path of length n in K_n. We would also like to mention here that the order of segments in the curve might not correspond to the order of vertices in the path.

Let Ψ be a fixed representation of a string graph G. A *shortest curve* in Ψ is a curve that is related to a shortest path between two distinct vertices in G. Let $P = u_1, \ldots, u_k$ be a shortest u_1, u_k-path and let C be a curve that is related to P. Let C be composed of segments $s_1, \ldots, s_l$ and the segments are in order $s_1, \ldots, s_l$. Then a segment of string $\psi(u_i)$ can only be adjacent to a segment of string $\psi(u_{i-1})$, $\psi(u_i)$, or of string $\psi(u_{i+1})$ in C (since P is a shortest path). Observe that, although multiple shortest curves can be related to a shortest path, a shortest curve relates to only one shortest path. For clarification of this observation, we present the following argument. Let $z_1, \ldots, z_k$ be natural numbers such that $z_1 = 1$, $z_k = l$, and $z_1 < z_2 < \cdots < z_k$. Then there exists a sequence $z_1, \ldots, z_k$ such that each segment $s \in \{s_{z_i}, \ldots, s_{z_{i+1}}\}$, for $1 \leq i \leq k-2$, is a segment of either the string $\psi(u_i)$ or the string $\psi(u_{i+1})$, and the segment $s_{z_{i+1}}$ is a segment of the string $\psi(u_{i+1})$. For $i = k-1$, each segment $s \in \{s_{z_i}, \ldots, s_{z_{i+1}}\}$, is a segment of either the string $\psi(u_i)$ or the string $\psi(u_{i+1})$. Thus, C can be related to only one path $u_1, u_2, \ldots, u_k$. Therefore, a shortest curve relates to only one shortest path.

A curve with endpoints a and b is referred to as an *a,b-curve*. Two curves are said to be *internally disjoint* if they can intersect only at their respective endpoints. Let C be a curve in a string representation Ψ. A curve π is said to be a *sub-curve of C* if π can be formed by some segments of C. We borrow the following topological lemmas by Gavenčiak et al. [56] that we will use in this chapter.

Lemma 4.2.1 (Gavenčiak et al. [56])**.** *Let B be a region. If π is a shortest curve and $\pi' \subseteq \pi$ is a sub-curve with $\pi' \subseteq B$, then π' is a shortest curve in Ψ_B.*

Lemma 4.2.2 (Gavenčiak et al. [56])**.** *Let π_1 and π_2 be two internally disjoint shortest a,b-curves with $a \neq b$ and F be one of the closed faces of $\mathbb{R}^2 \backslash (\pi_1 \cup \pi_2)$. For any simple a,b-curve π_3 contained in F and going through at least one of its inner points we have that every face of $\mathbb{R}^2 \backslash (\pi_1 \cup \pi_2 \cup \pi_3)$ is bounded by simple and internally disjoint curves π'_i and π'_3 with $\pi'_i \subseteq \pi_i$, $\pi'_3 \subseteq \pi_3$ and $i \in \{1, 2\}$.*

4.2.1 Brief Survey

Aigner and Fromme [3] proved that the cop number for the class of planar graphs is three. For that purpose, they proved that one cop can guard a shortest u,v-path after a finite number of steps. This guarding result and its extensions have been used extensively in computing the cop number of various graph classes, and we will also use this result . We here present a reworded version of the statement and its proof on the lines of the proof of the same by Bonato and Nowakowski [24].

Result 4.2.1 (Aigner and Fromme [3])**.** *Let u_0 and u_k be two distinct vertices of a graph G, and let P be a shortest u_0, u_k-path. Then one cop $\mathcal{C}$ can guard P after a finite number of moves.*

Proof. Let path $P = \{u_0, \ldots, u_k\}$. We define the *image* of the robber, denoted by $image(\mathcal{R})$, as u_i if $\mathcal{R}$ is at vertex v and $v \in D_i$. Hence $image(\mathcal{R})$ is always restricted to P and can only move from u_i to u_{i-1}, u_i or u_{i+1}, if they exist. Hence, $\mathcal{C}$ can occupy the vertex $image(\mathcal{R})$ in at most k moves. Thereafter, in whatever way $\mathcal{R}$ moves, after the move of $\mathcal{C}$, $\mathcal{C}$ can and will always continue to be on $image(\mathcal{R})$. Since each vertex $v \in P$ is an image of itself and after each round $\mathcal{C}$ is on $image(\mathcal{R})$, $\mathcal{R}$ gets captured by $\mathcal{C}$ whenever $\mathcal{R}$ enters P. Hence $\mathcal{C}$ guards P in at most k rounds. □

Using Result 4.2.1, Aigner and Fromme [3] proved that the cop number of planar graphs is three. We give a high level overview of their technique here. Consider a particular planar embedding of a planar graph G. Let the three cops be denoted as $\mathcal{C}_1, \mathcal{C}_2$, and $\mathcal{C}_3$. The cop $\mathcal{C}_1$ starts with guarding a shortest u, v-path P and then $\mathcal{C}_2$ guards a shortest u, v-path P' in $(G \setminus P) \cup \{u, v\}$. By Jordan Curve theorem, we can see that the paths P and P_1 divide the graph in two planar regions: one inside and one outside. Let those regions be A and B, respectively. Observe that in a planar graph, if $\mathcal{R}$ has to move from a vertex of A to a vertex of B, or vice versa, then $\mathcal{R}$ has to enter a vertex of $P \cup P_1$. Since both P and P_1 are guarded, $\mathcal{R}$ is restricted to one of the regions. Let this region be A. Now, $\mathcal{C}_3$ finds a shortest u, v-path P_2 in $A \cup \{u, v\}$ and guards it. Let the region enclosed between P and P_2 be A_1 and the region enclosed between P_2 and P_1 be A_2. Now, $\mathcal{R}$ is restricted in either the region A_1 or the region A_2, and depending on that we can free $\mathcal{C}_2$ or $\mathcal{C}_1$, respectively. Now the freed cop can again reduce the robber territory. This way cops go on reducing the robber territory and finally capture the robber. This result used the embedding structure of the planar graphs and this technique was later extended to other embeddable graphs as discussed below.

Let u and v be two distinct vertices of a graph $G(V, E)$. We say that a u, v-path P is a *shortest path relative to* $T \subseteq V$, if there is no shorter

u, v-path in $G|_{P \cup T}$ (graph induced by vertices of P and T). Gavenčiak et al. [56] showed that if P is a shortest path relative to $T \subseteq V$ and $\mathcal{R}$ is restricted to T, then one cop can guard P. They argued that the cop can guard P by considering the game in the graph induced by vertices of P and T. Here also, if the path P has length k, then the cop can guard P in at most k cop moves. They had the following lemma.

Lemma 4.2.3 (Gavenčiak et al [56])**.** *Let u and v be two distinct vertices of G and P be a shortest u, v-path relative to $T \subseteq V$, and $\mathcal{R}$ is restricted to T. Then one cop can guard P in a finite number of moves.*

Chiniforooshan [33] extended the idea of guarding a shortest path P to guarding $N[P]$, and used it to upper bound the cop number of a graph $G(V, E)$ in terms of $n = |V|$. This idea of guarding $N[P]$ was used by Gavenčiak et al. [56] to bound the cop number of string graphs. We will use this result for our algorithm and hence we give a small proof which is a rewording of Gavenčiak et al. [56].

Result 4.2.2 (Chiniforooshan [33])**.** *Let P be a shortest u_0, u_k-path in G, then five cops can guard $N[P]$ after a finite number of moves.*

Proof. Let $P = u_0, u_1, \ldots, u_k$. We refer to one of the five cops as the *sheriff* and the other cops are its *deputies*. The deputies follow the movements of the sheriff such that when the sheriff is at a vertex u_i, for $0 \leq i \leq k$, the deputies are at vertices $u_{i-2}, u_{i-1}, u_{i+1}$ and u_{i+2}. Let vertices u_{k+1} and u_{k+2} refer to the vertex u_k, and let vertices u_{-1} and u_{-2} refer to the vertex u_0.

Using Result 4.2.1, the sheriff guards P, in at most k steps. Next, we show that if the sheriff is guarding P, then the sheriff and its deputies are guarding $N[P]$. Suppose $\mathcal{R}$ moves to a vertex $v \in N[P]$, and let $v \in D_i$. So the sheriff has to move to u_i for guarding P, from a vertex $u \in \{u_{i-1}, u_i, u_{i+1}\}$. Observe that these five cops can ensure their presence at all these u_{i-1}, u_i, and u_{i+1} vertices, when the sheriff was at vertex u.

Also, since $v \in N[P]$, $v \in D_j$ and P is a shortest u_0, u_k-path, observe that v is adjacent to at least one of u_{i-1}, u_i, and u_{i+1}. Because there is a cop at each of these vertices, the cops will capture $\mathcal{R}$.

Thus five cops can guard $N[P]$, in at most k rounds. $\square$

Let T be a a subset of the vertex set of G. Gavenčiak et al [56] extended the idea of guarding the neighbourhood of a shortest path to guarding the neighbourhood of a shortest path relative to $T \subseteq V$. We state the lemma below. The proof is similar to the proof of Result 4.2.2.

Lemma 4.2.4 (Gavenčiak et al [56])**.** *Let u and v be two distinct vertices of G, P be a shortest u, v-path relative to $T \subseteq V$, and $\mathcal{R}$ is restricted to T. Then five cops can guard $N[P]$ after a finite number of moves.*

Using Lemma 4.2.4, Gavenčiak et al [56] proved that five cops can restrict the robber to cross a shortest path P in a string graph. Then using this result and some other sophisticated results, and a technique similar to that of Aigner and Fromme [3], they proved that 15 cops can always capture the robber in a string graph.

We improve upon this result and prove that 14 cops are always sufficient to capture the robber in a string graph. For that purpose, we prove that if P is the *unique shortest* u, v-path in G, that is, all other u, v-paths are longer than P, then four cops can guard $N[P]$.

This technique of restricting the robber to enter or cross a path was also used by Beveridge et al. [14] for bounding the cop number of unit disk graphs. They showed that three cops can restrict the robber to cross a shortest path in a unit disk graph. Then using this result and techniques similar to that of Aigner and Fromme [3], they proved that the cop number for the class of unit disk graph is at most 9.

4.3 Cops and Robber on String graphs

4.3.1 Guarding the unique shortest path

In the results discussed till now, five cops were guarding any shortest u, v-path and its neighbourhood, and then building on the techniques of Aigner and Fromme [3], one could reduce the robber territory. Our crucial observation is that if the shortest u, v-path is unique in some sense, then we can guard the path and its neighbourhood using four cops. Let G be a graph, and let u_0 and u_k be two distinct vertices of G. Then $P = u_0, \dots, u_k$ is *the unique shortest u_0, u_k-path*, if the length of every other u_0, u_k-path is more than the length of P. We will show that four cops are sufficient to guard $N[P]$.

Consider a graph $G(V, E)$. Let T be a subset of the vertex set V, that is, $T \subseteq V$. Consider a u_0, u_k-path P such that $P = u_0, u_1, \dots, u_k$. P is said to be a *special path relative to* $T \subseteq V$, if P is a shortest path relative to T and there is no vertex $v \in T$ such that $v \notin P$, $d(u_0, v) = i - 1$ and $v \in N(u_i)$, for $0 \leq i \leq k$.

Let u and v be two distinct vertices of G and let P be the unique shortest u, v-path. Then, observe that P is also a special path relative to $T = V$. Moreover, if a path P is a special path relative to $T \subseteq V$, then T is also a special path relative to T', where $T' \subseteq T$.

We now prove the following lemma, which is central to our proof. We use the notions of *sheriff* and *deputies* as in the proof of Result 4.2.2.

Lemma 4.3.1. *Let $P = u_0, \dots, u_k$ be a special path relative to $T \subseteq V$. Then 4 cops can guard $N[P]$, after a finite number of steps.*

Proof. We mark one cop as the *sheriff* and the other three cops are said to be its *deputies*. The deputies follow the movements of the sheriff such that when the sheriff is at a vertex u_i, for $0 \leq i \leq k$, the deputies are at vertices u_{i-2}, u_{i-1} and u_{i+1}. Let the vertex u_{k+1} refers to the vertex u_k, and let vertices u_{-1} and u_{-2} refer to the vertex u_0.

Since P is a shortest path relative to T, the sheriff guards P in at most k steps using Lemma 4.2.3. Moreover, it is worth mentioning that the sheriff can do so by staying on the vertices of P. More specifically, after each move of the sheriff, if $\mathcal{R}$ is at a vertex $v \in D_j$, then the sheriff is at vertex u_j. We claim that once the sheriff guards P, these four cops guard $N[P]$.

To prove the above claim, we show that if $\mathcal{R}$ moves to a vertex $x \in N[P]$ (also $x \in T$), then $\mathcal{R}$ gets captured by one of the cops. If $\mathcal{R}$ moves to a vertex in P, then the sheriff will capture the robber as it is guarding P. Let $\mathcal{R}$ moves to a vertex $x \notin P$, $x \in N[P]$, and $x \in D_j$.

Let $1 < j < k$. Since $x \in N[P]$, x is adjacent to at least one vertex of P. Now x cannot be adjacent to a vertex y from $\{u_0, \ldots, u_{j-2}\}$, as through path $u_0, \ldots, y, x$ the distance $d(u_0, x) < j$, which is not possible since $x \in D_j$. Moreover, x cannot be adjacent to a vertex y from $\{v_{j+2}, \ldots v_k\}$, as the path $u_0, \ldots x, y, \ldots u_k$ becomes a shorter u_0, u_k-path than P, which is a contradiction to the fact that P is a shortest path. Also, x cannot be adjacent to u_{j+1} by the definition of the special path. Hence, x can only be adjacent to u_{j-1} and u_j, and is adjacent to at least one of them. Since the sheriff is guarding $\mathcal{R}$, it can reach u_j in this cop move, and hence is at one of the vertex from $\{u_{j-1}, u_j, u_{j+1}\}$. In any case, there are cops on both u_j and u_{j-1}. Hence, one of these cops will capture $\mathcal{R}$ whenever R enters x.

Similar arguments hold for $j \in \{0, 1, k\}$. If $j = k$, then observe that x can only be adjacent to u_k and u_{k-1}, and both these vertices would be occupied by cops. If $j = 1$, then x can only be adjacent to u_0 and u_1, and both these vertices would be occupied by cops. If $j = 0$, then $x = u_0$ and hence x in on P, and since the sheriff is guarding P, it will capture $\mathcal{R}$.

Hence, these four cops can guard $N[P]$ in at most k steps. $\square$

If a curve π related to a special path P relative T, then π is referred to as a *special curve relative to* T. We extend Lemma 4.2.1 to accommodate the special curves in the following lemma.

Lemma 4.3.2. *Let B be a region of Ψ. If π is a special curve relative to T and $\pi' \subseteq \pi$ is a sub-curve with $\pi' \subseteq B$, then π' is a special curve relative to T in Ψ_B.*

Proof. First, we prove that π' is a special curve in Ψ. Let the curve π be related to a special u_0, u_k-path $P = u_0, \ldots, u_k$ in G. Then any sub-curve $\pi' \subseteq \pi$ would relate to a u_i, u_j-path $P' = u_i, u_{i+1}, \ldots, u_j$, where $0 \leq i \leq j \leq k$. For contradiction, let us assume that π' is not a special curve relative to T in Ψ, and hence P' is not a special path relative to T in G. Thus, there exists a vertex $v \in T \setminus P$ and some u_l (where $i \leq l \leq j$) such that $d(u_i, v) = d(u_i, u_l) - 1$ and $u_l \in N[v]$. Therefore, $d(u_0, u_i) + d(u_i, v) = d(u_0, u_i) + d(u_i, u_l) - 1$. Hence, we have a vertex $v \in T \setminus P$ such that $d(u_0, v) = d(u_0, u_l) - 1$ and $u_l \in N[v]$. This contradicts the fact that P is a special path (relative to T) related to special curve π (relative to T). Hence, π' is a special curve in Ψ and P' is a special path in G, both relative to T.

Next, we show that if a curve π' is a special curve in Ψ and $\pi' \subseteq \Psi_B$ (for some B of Ψ), then π' is a special curve in Ψ_B. Consider two vertices x and y of G corresponding to strings $\psi(x)$ and $\psi(y)$ in Ψ, respectively. Let x' and y' be two vertices in G_B such that $\psi(x')$ is a portion of $\psi(x)$ and $\psi(y')$ is a portion of $\psi(y)$. Then observe that $d(x', y')$ in G_B cannot be less than $d(x, y)$ in G.

Now consider a vertex v' in G_B such that $v' \notin P'$, corresponding to string $\psi(v')$ in Ψ_B, such that $u_l \in N[v']$. Let $\psi(v)$ be a string in Ψ, corresponding to vertex v such that $v \notin P$, such that $\psi(v')$ is a portion of string $\psi(v)$ in Ψ. Hence u_l is also a neighbour of v in G. Since P' is a special path in G, either $d(u_i, v) = d(u_i, u_l) + 1$ or $d(u_i, v) = d(u_i, u_l)$ in G. Hence, $d(u_i, v) \geq d(u_i, u_l)$ in G. Since $d(x', y')$ in G_B cannot be less than $d(x, y)$ in G, $d(u_i, v') \geq d(u_i, u_l)$. Thus, there cannot be any vertex

v' in $G_B \setminus P'$ such that $u_l \in N[v']$ and $d(u_i, v') = d(u_i, u_l) - 1$. Hence, P' is a special path in G_B and π' is a special curve in Ψ_B, both relative to T. $\square$

Also, if a curve π relates to a shortest path P relative to T, then we say that π is a *shortest curve relative to* T.

We would like to note that if a path P is a shortest/special relative to $T \subseteq V$, then P is shortest/special relative to every subset $T' \subseteq T$. Similarly, if a curve π is a shortest/special relative to $T \subseteq V$, then π is shortest/special relative to every subset $T' \subseteq T$.

4.3.2 Bounding the robber region

Consider a string representation of a string graph $G(V, E)$ in $\mathbb{R}^2$. Let a vertex $v \in V$ be represented by a string $\psi(v)$. Let $\Psi = \{\psi(v) \mid v \in V\}$ be the set of strings in our representation. We say that a string $\psi(v)$ is a *top-most string* if some point on $\psi(v)$ has the highest *y-coordinate* in Ψ. Here we also say that v is a *top-most vertex*. Similarly, we define the *bottom-most string* and *bottom-most vertex*.

Shortest paths represented by curves: Let C be a curve related to a path P. When some cops are guarding $N[P]$, then we also say that the cops are *guarding the curve* C.

Let u and v be two distinct vertices of G such that u is a top-most and v is a bottom-most vertex. Consider a shortest u, v-path P in G. Let p be a point on string $\psi(u)$ such that p has the highest *y-coordinate* in Ψ and p' be a point on $\psi(v)$ such that p' has the lowest *y-coordinate* in Ψ. Then a p, p'-curve C, related to a shortest path P, is referred to as a *top-bottom* curve. Note that this curve may not be unique.

Let C be a top-bottom curve related to a shortest u, v-path P such that u is a top-most and v is a bottom-most vertex. Observe that, if a vertex $x \notin N[P]$, then $\psi(x)$ lies either completely on the left of C or completely on the right of C. If a string $\psi(x)$ lies on the left (or right) of curve C, then we also say that vertex x lies on the left (or right) of P. We say that the robber *crosses* the curve C if $\mathcal{R}$ moves (in some finite rounds) from a vertex u completely on the left of C to a vertex v completely on the right of C, or vice versa.

This curve C is a continuous curve from a top-most point a to a bottom-most point b in the string representation. Hence, if a vertex x is on left of P and a vertex y is on right of P, every path from x to y passes through a vertex of $N[P]$. Thus, if cops are guarding $N[P]$, then the robber cannot *cross* any curve C corresponding to the path P. Here, we also say that the robber cannot *cross* the path P.

The following observation is a consequence of Result 4.2.2.

Observation 4.3.1. *Let u and v be two distinct vertices such that u is a top-most vertex and v is a bottom-most vertex, and P be a shortest u,v-path. If C is a curve related to path P, then five cops can restrict the robber to cross the curve C.*

Proof. Cops can achieve this by guarding $N[P]$. $\square$

Robber territory and restricted Graphs: Let $G(V,E)$ be a string graph. We say that $T \subseteq V$ is the *robber territory* if $\mathcal{R}$ cannot leave vertices of T without getting captured. Alternately, we can also say that $\mathcal{R}$ cannot move to a vertex $v \in V \setminus T$. We also say that $\mathcal{R}$ is *restricted* to T.

Consider a fixed string representation Ψ of G. We extend the definition of robber territory to *geometric robber territory*. Consider a region B. Let $\mathcal{R}$ be on a vertex u such that all points of the string $\psi(u)$ are inside the region B (points may be on the boundary of B also). If $\mathcal{R}$ cannot move to a vertex v (without getting captured) such that some portion of the string $\psi(v)$ is outside B, then we say that Ψ_B is the *geometric robber territory*. We say that $\mathcal{R}$ *is in the region* Ψ_B if $\mathcal{R}$ is on a vertex u of G such that all points of the string $\psi(u)$ are inside region B. Now we give various ways to bound the geometric robber territory.

We say that two curves are *internally disjoint* if they can intersect only at their respective endpoints. If a region B is bounded by two internally disjoint curves C_1 and C_2, then we denote Ψ_B as Ψ_{C_1,C_2} also. We have the following observation.

Observation 4.3.2. *Let π_1 and π_2 be two disjoint a, b-curves and $\mathcal{R}$ is in the region Ψ_{π_1,π_2}. If both curves π_1 and π_2 are guarded, then $\mathcal{R}$ cannot leave the region Ψ_{π_1,π_2} without getting captured.*

We extend this definition of bounding a region B with two curves C_1 and C_2 to bounding the region on the *left* or the *right* of a top-bottom curve C. Let C be a top-bottom curve. The region on the left of the curve C contains the points both on C and on left of C. Here Ψ_B is defined analogously and Ψ_B is also denoted by $\Psi_{C,L}$. Similarly, we can define the region on the right of the curve C, and also $\Psi_{C,R}$.

Observe that when we guard the closed neighbourhood of a shortest u, v-path P related to a top-bottom curve C, as we did in Observation 4.3.1, the robber territory T is restricted to either left or right of P.

Here we use curves to bound the robber territory and geometric robber territory. The following observations provide one more way to bound the robber territory.

Observation 4.3.3. *Let x be a cut vertex of $G(V, E)$ such that $G|_{V \setminus x}$ gives a connected component $G'(T, E')$. If $\mathcal{R}$ is on a vertex $v \in T$ and a cop is occupying the vertex x, then $\mathcal{R}$ is restricted to T and T becomes the robber territory.*

4.3.3 Extending a shortest path

Consider a string graph G and a fixed representation Ψ of G. Let Ψ_B be the geometric robber territory, and G_B be the graph corresponding to Ψ_B. Let Ψ_B be bounded by two internally disjoint a, b-curves π_1 and π_2, that is, Ψ_{π_1,π_2} is the geometric robber territory. Let π_1 and π_2 are related to paths P_1 and P_2 respectively. Also, let $T \subseteq V(G_B)$ be the robber territory such that both P_1 and P_2 are shortest paths relative to T. Intuitively speaking, by *extending a shortest path* P_1 (that relates to a shortest curve π_1), we mean that we find a shortest path P and a curve π such that in the region B' bounded by two disjoint curves $\pi_1' \subseteq \pi_1$ and $\pi' \subset \pi$, the curve π_1' is a special curve in the region $\Psi_{\pi_1',\pi'}$ and π' is a shortest curve in $\Psi_{\pi_1',\pi'}$. Here π_1' is a special curve in $\Psi_{\pi_1',\pi'}$ relative to $T' = V(G_{B'})$, and π' is a shortest curve in $\Psi_{\pi_1',\pi'}$ relative to $V(G_{B'} \setminus P_1')$, where P_1' is the path related to the curve π_1'. We can *extend the path* P_1 in the following manner.

If there is no u_0, u_k-path other than P_1 and P_2 in G_B, then we say that we *cannot extend the path* P_1. Otherwise, we do the following.

If P_1 is a special path in G_B relative to T, then we find a shortest u_0, u_k-path P in G_B other than P_1 and P_2. We call P as the *extended path* of P_1. Observe that P is a shortest path relative to the robber territory $T \setminus (P_1 \cup P_2)$ (and paths P_1 and P_2 are guarded), and hence $N[P]$ can be guarded using five cops. Let π be an a, b-curve related to path P. Since path P_1 is a special path in G_B, π_1 is a special curve. Hence, observe that if $\mathcal{R}$ is in a region bounded by two disjoint curves $\pi_1' \subseteq \pi_1$ and $\pi' \subseteq \pi$, then four cops can guard π_1' and five cops can guard π'. Here we need 9 cops to bound this robber region $\Psi_{\pi,\pi'}$.

If P_1 is not a special path in G_B relative to T, then we do the following. Let Q refer to path P_2. Find a vertex $x \in G_B \setminus (P_1 \cup Q)$ with least $d(u_0, x)$ such that $d(u_0, x) = i - 1$ and $u_i \in N(x)$, for $0 < i < k$ (there might be multiple such vertices and we select any one of them). Now consider the path $P = u_0, \ldots, x, u_i, \ldots, u_k$. We fix a curve C related to P in the following manner. Let p' be an intersection point of strings $\psi(x)$ and $\psi(u_i)$. Let C_1 be an a, p'-curve related to a shortest u_0, x-path which is not a subpath of the path P_1 or of the path P_2. Let p be a point on the string $\psi(u_i) \cap \pi_1$ which is closest to p' along the sub-curve of the string

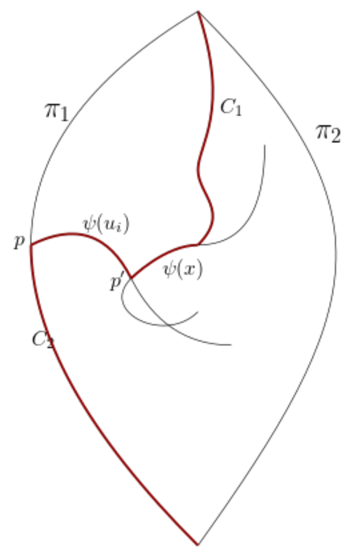


Figure 4.3.1: Here the extended path π is represented in red color.

$\psi(u_i)$. Let C_2 be the p, b-curve such that C_2 is a sub-curve of π_1. Then consider the curve π as the union of curves C_1 and p', p-curve that is a sub-curve of string $\psi(u_i)$ and C_2. See Figure 4.3.1 for an illustration. Note that P is a shortest path relative to $T \setminus (P_1 \cup P_2)$ and π is a shortest curve relative to $T \setminus (P_1 \cup P_2)$.

Now consider the region B' bounded by curves π_1 and π. If P_1 is a special path relative to $G_{B'}$, then we call P as the *extended path of* P_1. If not, then we consider the path P as Q, the region B' as B and repeat the steps of the above paragraph until we find an extended path P of P_1.

Finally, we get an extended path P of P_1. We also have curves π related to P and π_1 related to P_1. We call the curve π as the *extended curve of* π_1. Now, consider any region B' bounded by two disjoint curves $\pi'_1 \subseteq \pi_1$ and $\pi' \subset \pi$, and let $\mathcal{R}$ be inside the region $\Psi_{\pi'_1,\pi'}$. The curve π' is a special curve in the region $\Psi_{\pi'_1,\pi'}$ relative to $T' = V(G_{B'})$. The curve π'_1 is a shortest curve in region $\Psi_{\pi'_1,\pi'}$ relative to $V(G_{B'} \setminus P'_1)$, where P'_1 is a path related to curve π'_1. Now we can guard π using four cops, and when π' is guarded (or even P' is guarded), we can guard π'_1 using five cops. Hence, the geometric robber territory $\Psi_{\pi'_1,\pi'}$ can be bounded using 9 cops.

If Ψ_B is bounded by only one top-down curve π_1 related to path P_1, that is, the robber territory is either $\Psi_{\pi_1,L}$ or $\Psi_{\pi_1,R}$, then we extend the path P_1 in following manner. We use a similar strategy to the above one, with the only change that we consider the path P_2 as an empty path, that is, $P_2 = \emptyset$. Rest we follow the same steps and find an extended path P of P_1.

We have the following observation.

Observation 4.3.4. *Let P_1 be a shortest path in G_B (relative to some $T \subseteq V(G_B)$) and P is an extended path of P_1. Moreover, let the curves π_1 and π be related to paths P_1 and P, respectively such that π is the extended curve of π_1. Let both π_1 and π are guarded by cops. Consider a region B' bounded by two disjoint curves $\pi'_1 \subseteq \pi_1$ and $\pi' \subseteq \pi$. The curve π' is a special curve in the region $\Psi_{\pi'_1,\pi'}$ relative to $T' = V(G_{B'})$. The curve π'_1 is a shortest curve in region $\Psi_{\pi'_1,\pi'}$ relative to $V(G_{B'} \setminus P'_1)$, where P'_1 is the path related to the curve π'_1. Thus, if $\mathcal{R}$ is inside the region B', then 4 cops can guard π' and 5 cops can guard π'_1. Hence, 9 cops can bound the geometric robber territory $\Psi_{\pi'_1,\pi'}$.*

We use this technique of extending a path to ensure that, in our algorithm, whenever two teams of cops guard a bounded region, they can do so using 9 cops instead of 10 cops (as done by Gavenčiak et al. [56]) giving us a maximum cop number of 14 instead of 15. Now we are ready to present our algorithm and give our algorithm in the next section.

4.3.4 Algorithm

Now we show that 14 cops are always sufficient to capture the robber in a string graph G. Let $G(V, E)$ be a string graph and Ψ be a fixed representation of G. Let $\mathcal{R}$ is in a region B and $\mathcal{R}$ is restricted to $T \subseteq$

$V(G_B)$. Let u and v be two distinct vertices in G_B. For our strategy, first we define three game states, *state 1*, *state 2* and *state 3* as follows.

1. *State 1*: Let u be a top-most and v be a bottom-most vertex in G_B. Then five cops are guarding a shortest u, v-path P.

2. *State 2*: The region B is bounded by two disjoint curves π_1 and π_2 such that π_1 is a special curve in Ψ_{π_1,π_2} relative to T and π_2 is a shortest curve in Ψ_{π_1,π_2} relative to T. Then five cops are guarding π_2 and four cops are guarding π_1 (total 9 cops).

3. *State 3*: Let x be a vertex in G such that G_B is a connected component of $G \setminus x$. If a cop is occupying the vertex x and $\mathcal{R}$ is in G_B, then observe that $\mathcal{R}$ is restricted to G_B. Let u be a top-most vertex and v be a bottom-most vertex in G_B, and P be a shortest u, v-path relative to T. Then one cop is occupying vertex x and five cops are guarding guard $N[P]$. Moreover, $\mathcal{R}$ and $\psi(x)$ are on the same side of each curve π related to P.

State 1, State 2, and State 3 are referred to as *safe states*. (Observe that, in safe state 1, the robber is restricted either to the left or to the right of P.)

We have the following lemma which is central to our algorithm.

Lemma 4.3.3. *Consider a fixed representation Ψ of a string graph $G(V, E)$. Let $\mathcal{R}$ be in a region B of Ψ, and Ψ_B be the geometric robber territory. Let $T \subseteq V(G_B)$ be the robber territory, and the game is in a safe state S. Then 14 cops can force the game to a safe state S' and robber territory to $T' \subseteq T$ and geometric robber territory to $\Psi_{B'} \subset \Psi_B$, in a finite number of moves.*

Proof. Depending upon the state S of the game, we do the following:

1. **S = state 1**: Let u_0 be a top-most and u_k be a bottom most vertex in G_B , and $P = u_0, \ldots, u_k$ be the shortest path such that $N[P]$ is

guarded by 5 cops. Let π be a curve related to path P (and let π defined B). Observe that the geometric robber territory is either $\Psi_{\pi,L}$ or $\Psi_{\pi,R}$. Without loss of generality, let us assume that $\mathcal{R}$ is restricted to the right of π and hence $\Psi_{\pi,R}$ is the geometric robber territory. We extend the path P in $\Psi_{\pi,R}$ and let P' be the extended path of P. Now, guard $N[P']$ using 5 cops. (We can do so because P' is a shortest path relative to $T \setminus P$.) Consider a top-bottom curve π' related to path P', such that π' is an extended curve of π. Now, one of the following scenarios is possible:

(a) Path P cannot be extended. It is possible only if there is no u, v-path in G_B other than P. Let $\mathcal{R}$ be in a connected component $G'(T', E')$ of $G_B \setminus P$. In this case, we claim that there is a unique vertex $x \in P$ such that x has a neighbour in G'. For contradiction, assume that there is some other vertex $y \neq x$ in P such that y has some neighbour in G'. Then consider the path P' formed by the vertices of u, x-path along P, followed by a shortest x, y-path in $G' \cup \{u, v\}$, followed by the y, v-path along P. Here P' is an extended path of P, and thus we have a contradiction. Thus x is a cut vertex such that $G_B \setminus x$ gives G' as a component.

    Guard x using one cop and free other cops. Now, find a top-most vertex u' and a bottom-most vertex v' in G' and a shortest u', v'-path in G'. Now, consider a top-down curve C' corresponding to path P' and guard C' using five cops. If $\mathcal{R}$ and x are on same side of P', then we are in the safe state 3. If $\mathcal{R}$ and x are on opposite sides, then we can free cop on x and we are in safe state 1. In both cases, at least the segments corresponding to vertices of $P - x$ will be removed from the geometric robber territory.

(b) $\mathcal{R}$ is not in the region bounded by curves π and π'. Observe that, here $\mathcal{R}$ is on the same side of π and π'. Since π' is a top-bottom curve and π' is guarded by five cops, we can free the cops on curve π. Hence, the geometric robber territory is now $\Psi_{\pi',R}$ (since $\mathcal{R}$ is in right of both π and π'). Also, $\Psi_{\pi',R} \subset \Psi_{\pi,R}$ since the region bounded between π and π' is in $\Psi_{\pi,R}$ but not in $\Psi_{\pi',R}$.

(c) $\mathcal{R}$ is in the region bounded by two curves C and C' such that $C \subseteq \pi$ and $C' \subseteq \pi'$. By Observation 4.3.4 we know that C is a special curve and C' is a shortest curve, both relative to new robber territory T'. Hence 4 cops can guard the curve C and 5 cops can guard C'. Hence we are in the safe state 2. For the sake of convenience, to prove that the geometric robber territory decreases in this case, we prove it for state 2, and whenever this case occurs, we execute this Lemma again for state 2.

2. **S = state 2**: Let B is bounded by two disjoint curves π and π', and $\Psi_{\pi,\pi'}$ is the geometric robber territory. Let π and π' are related to paths P and P', respectively. Also, let π be a special curve in $\Psi_{\pi,\pi'}$ and π' be a shortest curve in $\Psi_{\pi,\pi'}$, both relative to T. Let the curves π and π' intersect at points a and b. Also, four cops are guarding π and five cops are guarding π'. Now extend the path P' and let P_1 be the extended path of P'. Also let π_1 be a curve related to P_1 such that π_1 is an extended curve of π'. Now, guard π_1 using five cops. By Lemma 4.2.2, we know that each face of $\Psi_{\pi,\pi'} \setminus (\pi \cup \pi' \cup \pi_1)$ is bounded by simple and disjoint curves C_1 and C_2 such that $C_1 \subseteq \pi_1$ and, either $C_2 \subseteq \pi$ or $C_2 \subseteq \pi'$. If $\mathcal{R}$ is trapped between $C_1 \subseteq \pi_1$ and $C_2 \subseteq \pi$, then observe that C_1 is a shortest curve relative to the new robber territory T' and C_2 is a special curve relative to T' (by Observation 4.3.4). If $\mathcal{R}$ is trapped between $C_1 \subseteq \pi_1$ and $C_2 \subseteq \pi'$, then observe that C_1 is a shortest

curve relative to T' and curve C_2 is a special curve relative to T' (by Observation 4.3.4). In either case, we can guard C_1 using 5 cops and C_2 using 4 cops, and we are in safe state 2. Also, since C_2 is not equal to π or π', at least some part of π or π' is removed from the geometric random territory. Note that, we use 14 cops to reduce the robber territory here. (Any improvement in this step will reduce the cop number further).

Suppose we cannot extend the path P' (that is, there is no u, v-path in T other than P and P', where u and v are endpoints of P'). Then observe that the vertices of the connected component of $G_B \setminus (P \cup P')$ containing $\mathcal{R}$ can be connected to only one vertex x of $P \cup P'$ (Proof is similar to the argument in case 1(a)). We move one cop to vertex x and free all other cops. Now, we are in a situation similar to that of step 1(a). Hence we follow the same steps. Note that, we reduce the geometric territory of $\mathcal{R}$ in this step.

3. **S = state 3**: Let x be a vertex such that G_B is a connected component of $G \setminus x$. Consider the representation $\Psi' \subset \Psi$ such that $\Psi' = \{\psi(u) \mid u \in G_B\}$. Let u and v be a top-most and bottom-most vertex of G_B, respectively. Also P is a shortest u, v-path such that $N[P]$ is guarded by 5 cops, and one cop is occupying the vertex x. Moreover, both $\mathcal{R}$ and x are on the same side of P. Without loss of generality, let us assume that they are on the right of P. Since x is occupied by a cop and $N[P]$ is guarded by cops, observe that the geometric robber territory is $\Psi'_{\pi,R}$, where π is a curve related to P. Now, we extend the path P in $\Psi'_{\pi,R}$ and let P_1 be the extended path of P. Also let π_1 be related to P_1 such that π_1 is an extended curve of π in $\Psi'_{\pi,R}$.

   If $\mathcal{R}$ and x are on the same side of π_1, then we can free cops from P and we are in the safe state 3. Here, the geometric robber territory is reduced by region bounded between π and π_1.

If $\mathcal{R}$ is in the region bounded by π and π_1, then we are in the safe state 2. Here also, similar to case 1(a), we execute this Lemma again for the safe state 2, and we proved that the geometric territory decreases when we execute this Lemma starting from state 2.

If the path P cannot be extended in $\Psi'_{\pi,R}$, then there exists a vertex $y \in P$ such that vertices in $G_B \setminus y$ gives a connected component $G_{B'}$ containing $\mathcal{R}$. If $x \notin G_{B'}$, then we are in a situation similar to 1(a) and we follow the same steps. If $x \in G_{B'}$, then We place one cop on y and free other cops from P. Now, we find a top-most vertex u' and bottom-most vertex v' in $G_{B'}$ and find a shortest u', v'-path P_1 in $G_{B'}$. Now, five cops guard $N[P']$. Now consider a top-bottom curve π' related to P'. Now, either x and $\mathcal{R}$ lie on the same side of π' or y and $\mathcal{R}$ lie on the same side of π'. In both cases, we are in the safe state 3. Also observe that each segment s such that s is a segment of path P and s is not a segment of string $\psi(y)$ is reduced from the geometric robber territory. Hence the geometric robber territory reduces in this step.

This completes the proof of our lemma. $\square$

Now we prove the main theorem of this section.

Theorem 4.3.1. *If G is a string graph, then $c(G) \leq 14$.*

Proof. We give a cop strategy to prove our claim. We first show that at most 14 cops can force the robber to a safe state.

Initially, the robber territory $T = G$ and $G_B = G$. Cops find a top-most vertex u and a bottom-most vertex v in G_B and find a shortest u, v-path P in G_B. Now, five cops guard P. This restricts the robber either to the left or to the right of P. Now we are in the safe state 1.

After this, until the robber is captured, we use Lemma 4.3.3 to reduce the geometric robber territory. Since we have a finite graph with a finite representation and cops can reduce the geometric robber territory in every iteration of Lemma 4.3.3 using at most 14 cops, these 14 cops will eventually capture the robber. □

Gavenčiak et al. [56] showed that if a string graph G have girth 5 and cop number k, then G is k-degenerate. Using this result along with the Theorem 4.3.1 gives us the following corollary.

Corollary 5. *If G is a string graph with girth 5, then G is 14-degenerate and hence 15-colorable.*

Gavenčiak et al. [56] also proved that for a string graph G orientable on a surface of genus g, $10g + 15$ cops are sufficient to capture the robber. For this purpose, they use 10 cops to unfold a genus, and then finally capture the robber in a genus 0 string graph using 15 cops. If we use our strategy to capture the robber in a string graph of genus 0 using 14 cops, along with their unfolding techniques, we have the following immediate corollary.

Corollary 6. *If G is a string graph having genus g, then $c(G) \leq 10g+14$.*

4.4 Boxicity 2 graphs

Boxicity 2 graphs are the intersection graphs of axis-parallel boxes (rectangles) in $\mathbb{R}^2$. Let *2-BOX* be the family of boxicity 2 graphs. Since boxes are arc-connected, 2-BOX is a subset of the class of string graphs. Hence, we have the following corollary.

Corollary 7. *Let 2-BOX be the family of boxicity 2 graphs. Then $c(2\text{-}BOX) \leq 14$.*

Gavenčiak et al. [56] showed that $2 \leq c(\textit{2-BOX}) \leq 15$. We improve both the lower bound and upper bound for this result in the following theorem.

Theorem 4.4.1. *Let 2-BOX be the family of rectangle intersection graphs. Then* $3 \leq$*c(2-BOX)*≤ 14

Proof. Corollary 7 proves the upper bound. To prove the lower bound, $3 \leq$*c(2-BOX)*, we give a rectangle intersection representation of the dodecahedron graph in Figure 4.4.1. It is known from Aigner and Fromme [3] that the cop number of the dodecahedron graph is 3. Hence, $3 \leq$*c(2-BOX)*≤ 14. □

4.5 Concluding remarks and open problems

It will be interesting to see whether the techniques used in this chapter can be used to reduce the known upper bound on cop number from 9 ([14]) to 8. Gavenčiak et al. [56] showed that for a string graph G drawn on a surface with genus g, $c(G) \leq 10g + 15$. Using the techniques used in this chapter, this result can be improved to $c(G) \leq 10g + 14$. Gavenčiak et al. [56] unfold a genus g surface to genus $g - 1$ surface by guarding 2 shortest paths using 10 cops. It would be interesting to see if similar techniques could be used to unfold using 9 cops as that can give us $c(G) \leq 9g + 15$.

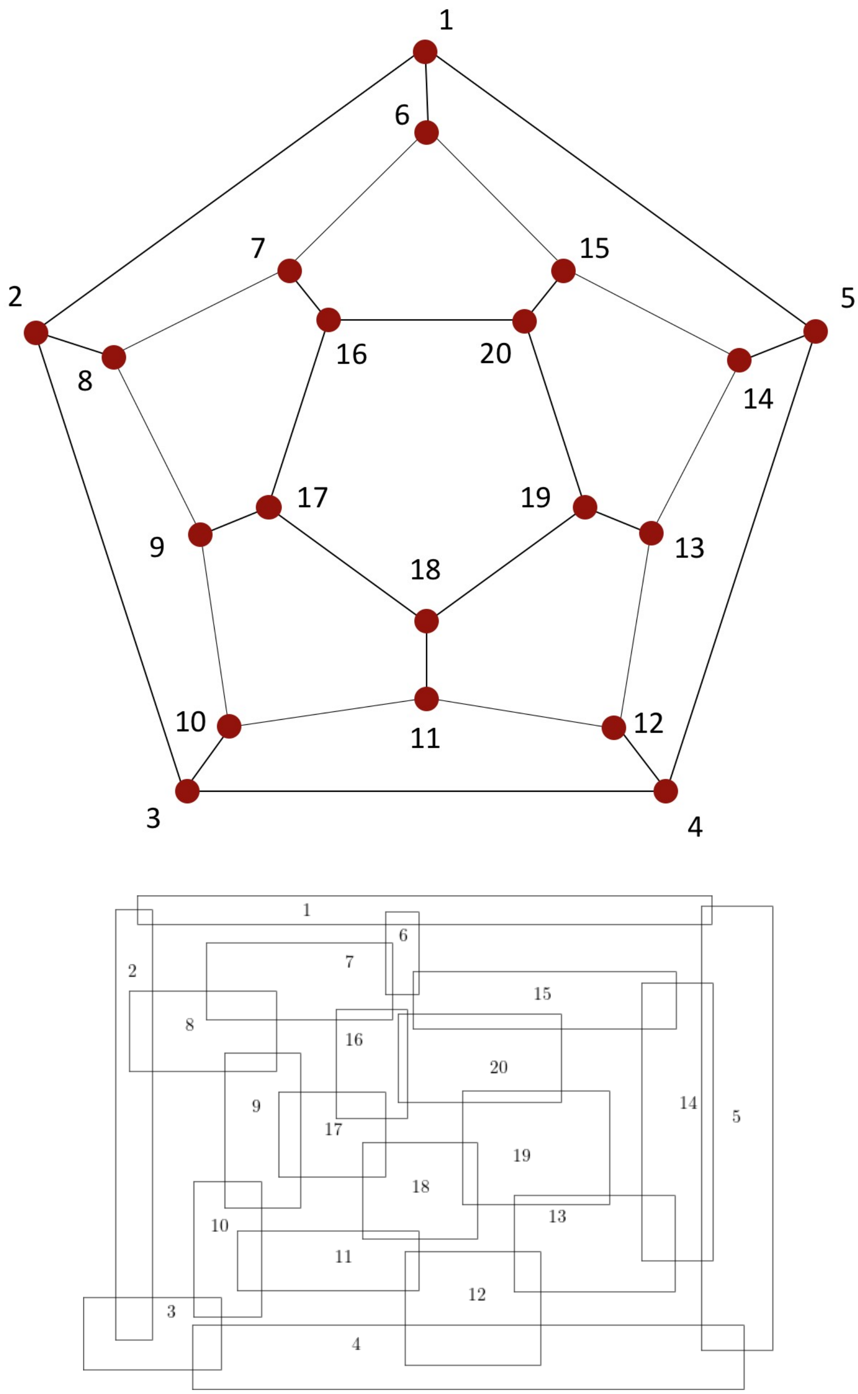


Figure 4.4.1: A dodecahedron and its boxicity 2 representation. Here each vertex i corresponds to the rectangle i.

5

Applications of Guarding Subgraphs

Contents

Let H be a subgraph of G. We say that cop(s) guard H if $\mathcal{R}$ cannot enter H without getting captured. The idea of guarding subgraphs is used heavily in Cops and Robber games literature to give strategies to capture the robber. In this chapter, we consider some applications of guarding subgraphs to the game of Cops and Robber. In particular, we use guarding techniques to find the cop number of butterfly graphs and AT-free graphs. We also study the game of Cops and fast Robber on graphs having a dominating pair.

5.1 Preliminaries

Let H be a subgraph of G. We say that $\mathcal{R}$ is *restricted* to H, if $\mathcal{R}$ cannot leave the vertices of H without getting captured. We also say that H is the *robber territory*.

Let H be a subgraph of G. A cop $\mathcal{C}$ *guards* H if the robber cannot enter the vertices of H without getting captured by $\mathcal{C}$ in the next cop move.

For a graph G, *capture time* using k cops, is the number of cop moves to ensure the capture (of robber) using k cops.

A k-dimensional butterfly network, is a graph, consisting of $2^k(k+1)$ vertices arranged in $k+1$ columns and 2^k rows. The 2^k rows are coded in k-bit binary from $00\ldots00$ to $11\ldots11$ and the $k+1$ columns are coded in decimal from 0 to k. These columns are also referred to as *levels*. A vertex in i-th row and j-th column is denoted by (i,j). There exists an edge between two vertices (i,j) and (i',j') if (1) $j' = j+1$ and (2) either $i' = i$, referred as *straight edge*, or the binary representations of i and i' differ exactly in the j'-th least significant bit, referred as *cross edge*. See Fig. 5.1.1 for an illustration.

We also define the *recursive* definition of a k-dimensional butterfly network, which we will use in our algorithm. For that purpose, take two $(k-1)$-dimensional butterfly networks A and B. Now, for all vertices of A, append 0 as the most significant bit and for all the vertices of B,

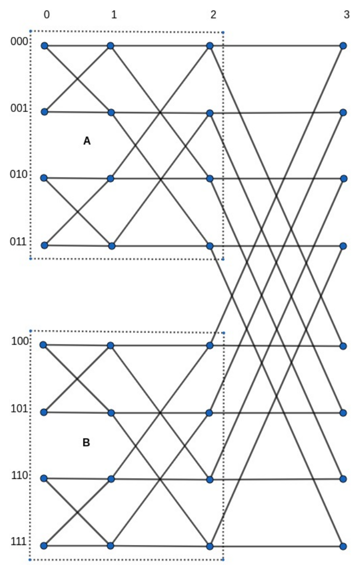


Figure 5.1.1: A 3-dimensional butterfly network.

append 1 as the most significant bit. Levels of all vertices remain as they were in A and B. Next, add level k with 2^k vertices and add edges between vertices of level k and $k-1$ as per the rules defined for butterfly networks in the previous definition. See Fig. 5.1.1 for an illustration. Observe that all paths from vertices of A to vertices of B go through the vertices of the new level (level k).

Butterfly networks are extensively studied interconnection networks and have applications in parallel computing [76]. We study the game of cops and robber on butterfly networks and have the following theorem.

Let u and v be two vertices of a graph G. A vertex u is a *corner vertex*, if there exists a vertex v such that $N[u] \subseteq N[v]$. We also say that u is a corner of v. If a graph G has no corner vertex, then $c(G) > 1$ [3, 86]. Aigner and Fromme [3] proved the following result (we restate the result to suit our definitions), which we will use.

Result 5.1.1. *(Aigner and Fromme [3]) Let P be a shortest path between two vertices u and v of a graph G. Then one cop can guard P after a* *finite number of moves.*

5.2 Chapter Overview

In Section 5.1 we define some important definitions and tools that we will use for our proofs. In section 5.3 we study the Cops and Robber game on butterfly networks and using a nontrivial and novel guarding technique, we prove that the cop number for the class of butterfly network graphs is 2. In section 5.4, we study the cop number for the class of AT-free graphs and improve the known bounds by Fomin et al. [46] for both the classical Cops and Robber game and for the Cops and fast Robber game.

In section 5.6 we suggest some future directions.

5.3 Butterfly Networks

In this section, we give a strategy to capture the robber in a k-dimensional butterfly network using two cops in $O(k^2)$ moves. We need the following definitions.

We refer the vertex in level 0 and row $00\ldots00$ as the *start vertex* of a k-dimensional butterfly network. An agent, cop or robber, makes a *forward move* if the level of vertex occupied by the agent increases, and makes a *backward move* if the level of the vertex occupied by the agent decreases.

When we say l-th least significant bits or l significant bits, it means the usual for $l > 0$. For $l = 0$, assume that the l least significant bits or the l-th least significant bit of all the tuples are the same.

Vertex (i', j) is said to be an *image* of (i, j), if the j least significant bits of i and i' are the same. For example, each vertex is an image of itself, all the vertices of level 0 are images of each other, and the vertices of level k do not have any image other than themselves. A cop $\mathcal{C}$ *captures an image* of $\mathcal{R}$ if $\mathcal{R}$ is at a vertex (i, j) and $\mathcal{C}$ is at an image of (i, j).

A path is a *monotone path* if all its vertices are from different levels and the first vertex is from level 0. A cop $\mathcal{C}$ *guards* level l if the robber cannot enter the vertices of level l without getting captured by $\mathcal{C}$ in the subsequent cop move.

Recall that we refer to $(00\ldots00, 0)$ as the start vertex. We have the following lemma.

Lemma 5.3.1. *Let (i, j) be a vertex of a butterfly network. Then there exists an image (i', j) of (i, j) such that there exists a monotone path from the start vertex to (i', j).*

Proof. We follow a simple strategy to create such a monotone path P. We start our path P from the *start vertex*. When we move from level l to $l + 1$, we take a straight edge if the $l + 1$-th least significant bit of i is 0, and take a cross edge otherwise. This way, when we reach the level j, at a vertex (i', j), the j least significant bits of i and i' will be the same, and each vertex of path P is in a different level. Hence this path P is a monotone path from start to an image (i', j) of vertex (i, j). $\square$

Lemma 5.3.2. *In a k-dimensional butterfly network, one cop can capture an image of the robber in at most k steps.*

Proof. Cop $\mathcal{C}$ begins at the start vertex. Let $\mathcal{R}$ be at a vertex (i, j). First, $\mathcal{C}$ will find a monotone path P from the start vertex to an image of $\mathcal{R}$ (by Lemma 5.3.1). The cop will update the monotone path P dynamically, following the moves of $\mathcal{R}$, and will move forward through P until it reaches an image of $\mathcal{R}$.

Let $\mathcal{R}$ be at vertex (r, l). We will maintain the invariant that the last vertex of P is (p, l) such that (p, l) is an image of $\mathcal{R}$. Also in each cop move, $\mathcal{C}$ will make a forward move on P. If at any point $\mathcal{C}$ and $\mathcal{R}$ are at the same level, then observe that $\mathcal{C}$ has captured an image of $\mathcal{R}$.

If $\mathcal{R}$ moves forward to increase the level from vertex (r, l) to $(r', l+1)$, then r and r' differ in at most one bit (that is the $(l+1)$-th least significant bit). Before this move, let the last vertex of P be (p, l). Since l least significant bits of p and r are the same, l least significant bits of p and r' are also the same. Thus, if the $l+1$-th bit of p and r' is the same, then we extend our path P using a straight edge, else we extend P using a cross edge.

If $\mathcal{R}$ moves backward, then we truncate our path by one vertex. Suppose $\mathcal{R}$ moves from (r, l) to $(r', l-1)$. Here r and r' differ in at most one bit, that is, the l-th bit. Hence the first $l-1$ bits of r, r', p and p's neighbour in level $l-1$ are the same. Therefore, our invariant holds if we just remove the last vertex from our path P.

Since in each cop move $\mathcal{C}$ is strictly increasing its level in a monotone path, in at most k moves, both $\mathcal{C}$ and $\mathcal{R}$ will be in the same level. Hence, $\mathcal{C}$ captures an image of $\mathcal{R}$ in at most k steps. □

Lemma 5.3.3. *In a k-dimensional butterfly network, one cop can guard level k in at most k steps.*

Proof. In the level k of a k-dimensional butterfly network, each vertex has only itself as an image. Thus, if $\mathcal{R}$ is in level k and $\mathcal{C}$ captures an image of $\mathcal{R}$, then $\mathcal{C}$ captures $\mathcal{R}$. Hence, if $\mathcal{C}$ can ensure that after each cop move $\mathcal{C}$ has captured an image of the robber $\mathcal{R}$, then $\mathcal{R}$ cannot enter level k without being captured by $\mathcal{C}$.

Cop $\mathcal{C}$ starts by capturing an image of the robber (by Lemma 5.3.2). In Lemma 5.3.2 we are maintaining a dynamic path P such that its last vertex (p, l) is an image of $\mathcal{R}$. When $\mathcal{C}$ captures an image of $\mathcal{R}$, cop $\mathcal{C}$ is on the last vertex (p, l) of P. We keep maintaining this dynamic path P as we did in Lemma 5.3.2, and $\mathcal{C}$ will move such that $\mathcal{C}$ is on the last vertex of P. Note that $\mathcal{C}$ may have to move backward (from a column c to column $c-1$) also. Let (p, l) be the last vertex of P and after the move of $\mathcal{R}$, the new last vertex of P is (p', l'). Since vertices (p, l) and (p', l') are adjacent, $\mathcal{C}$ can and will move to (p', l').

Thus, once $\mathcal{C}$ captures an image of $\mathcal{R}$ in a k-dimensional butterfly network, $\mathcal{R}$ cannot enter level k. Hence $\mathcal{C}$ guards level k when $\mathcal{C}$ captures an image of $\mathcal{R}$. $\square$

The following lemma is central to our strategy to capture $\mathcal{R}$ using two cops.

Lemma 5.3.4. *Let $\mathcal{R}$ be restricted to levels from 0 to x, for $1 \leq x \leq k$, in a k-dimensional butterfly network. Then after a finite number of moves, one cop, say $\mathcal{C}$, can restrict the robber to levels from 0 to $x-1$.*

Proof. If $x = k$, then $\mathcal{C}$ can restrict $\mathcal{R}$ to levels from 0 to $x-1$ simply by guarding level k (using Lemma 5.3.3).

If $x < k$, then we consider the recursive definition of butterfly networks. If we consider the levels from 0 to x of a k-dimensional butterfly network and consider only the x least significant bits of binary codes of rows, then we have 2^{k-x} butterfly networks of dimension x. (If $v = (i, j)$ was a vertex of original network, then here we consider the vertex v as (i', j), where i' is a x bit binary tuple containing x least significant bits of i.)

Now observe that, if $\mathcal{R}$ on a vertex of one of these x-dimensional butterfly networks, say A, then $\mathcal{R}$ cannot leave A without entering the level $x+1$ (as all these x-dimensional butterfly networks are connected only through vertices of level $x+1$). Now $\mathcal{C}$ will consider only the x least significant bits of the butterfly network A and follow the strategy from Lemma 5.3.3 to guard level x (here the start vertex becomes the start vertex of A). Once $\mathcal{C}$ guards level x, the robber $\mathcal{R}$ cannot enter level x and hence is restricted to levels from 0 to $x-1$. $\square$

Now we are ready to prove the main result of this section. In the following theorem, we prove that the cop number for a finite butterfly network is two.

Theorem 5.3.1. *Cop number for finite butterfly networks is two.*

Proof. We give a cop strategy to capture $\mathcal{R}$ in a k-dimensional butterfly network using two cops. In this strategy, cops keep restricting the robber territory level by level, finally restricting $\mathcal{R}$ to level 0, where it cannot move. Then the cops capture $\mathcal{R}$.

Initially, $\mathcal{R}$ is restricted to levels from 0 to k. Using Lemma 5.3.4, one cop restricts $\mathcal{R}$ to levels from 0 to $k-1$. While this cop guards $\mathcal{R}$, other cop moves and restricts $\mathcal{R}$ to levels from 0 to $k-2$ using Lemma 5.3.4. Once $\mathcal{R}$ is restricted to levels from 0 to $k-2$ by the second cop, the first cop guarding level k can be freed. (Cops can do so because if $\mathcal{R}$ cannot enter level $k-1$, it cannot enter level k.) This way whenever cops restrict $\mathcal{R}$ using a new cop, the previous cop gets free and restricts $\mathcal{R}$ further to smaller levels.

The cops, subsequently, restrict $\mathcal{R}$ to level 0 where one cop is ensuring the guard position. Now the second cop moves and captures $\mathcal{R}$. Hence, two cops are sufficient to capture the robber in a butterfly network.

To show that two cops are necessary, we prove a stronger result that all k-dimensional butterfly networks, for $k > 0$, have cop number greater than 1. We prove this by proving that k-dimensional butterfly networks, for $k > 0$, do not have a corner vertex. For contradiction, suppose that u and v are two vertices of a k-dimensional butterfly network such that u is a corner of v; so $N[u] \subseteq N[v]$. Thus, u and v must be adjacent and hence must be in different but consecutive levels. Now u has two neighbours in the level of v and one of them is v. Let the other neighbour be x. If $N[u] \subset N[v]$, then $x \in N[v]$. This is a contradiction as x and v are in the same level. Therefore, there is no corner vertex in a butterfly network. Thus, two cops are necessary to capture a robber in a butterfly network.

Hence, the cop number for butterfly networks is 2. $\square$

Since the cops capture $\mathcal{R}$ by restricting $\mathcal{R}$ to smaller levels in each iteration and each iteration takes $O(k)$ time for a k-dimensional butterfly network (having $2^k(k+1)$ vertices), we have the following corollary.

Corollary 8. *Capture time for a k-dimensional butterfly network using two cops is $O(k^2)$.*

5.4 Cops and fast Robber on AT-free graphs

In this section, we consider the game of Cops and fast Robber on AT-free graphs. For the game of Cops and fast Robber, the *Cop number* of a graph G, denoted by $c_s(G)$, is the minimum number of cops that are sufficient to capture the robber with speed s. If the speed of the robber is 1, then the game is equivalent to the classical Cops and Robber game.

Three independent vertices of a graph form an *asteroidal triple* if each pair of vertices is joined by a path that avoids the neighbourhood of the third vertex. A graph is *asteroidal triple-free* (AT-free) if it contains no asteroidal triple. Corneil et al. [39] showed that every connected AT-free graph contains a *dominating pair*, that is, a pair of vertices such that every path joining them is a dominating set in the graph.

Nisse and Suchan [85] studied the game where the robber is twice as fast as the cops and showed that the cop number of planar grids is unbounded for this game. Balister et al. [10] studied the game of Cops and fast Robber on grids where the robber can be arbitrarily fast. Fomin et al. [46] considered the game where the robber can move faster than cops and proved that, for a graph G having a dominating pair of vertices (hence also for AT-free graphs), $c_s(G) \leq 5s - 1$. We improve this bound and prove that for a graph G having a dominating pair, $c_s(G) \leq s + 3$.

5.4.1 Guarding a shortest path

Consider a graph G and let u and v be two distinct vertices of G. We say that a cop $\mathcal{C}$ *guards* a shortest u, v-path P, if $\mathcal{R}$ cannot enter this path without getting captured by $\mathcal{C}$. Aigner and Fromme [3] proved that for any two vertices u and v of G, one cop can guard a shortest u, v-path after a finite number of moves. Let $d(x, y)$ denote the shortest graph distance between two vertices x and y. Let r denote the vertex occupied by the robber. Recall that in Result 4.2.1, we showed that one cop can guard a shortest u, v-path P using the following strategy. $\mathcal{C}$ moves in such a manner that after each cop move, if $d(u, r) \leq d(u, v)$, then $\mathcal{C}$ is on a vertex x such that $d(u, x) = d(u, r)$, else C is on vertex v.

5.4.2 Algorithm

Let G be a graph having (u, v) as a dominating pair. Let P be a shortest u, v-path having length $k + 1$, that is, $d(u, v) = k + 1$. For the sake of convenience, we rename the vertices in the path P such that $P = u_0, u_1, \ldots, u_k$. For $i > 0$, vertices u_{k+i} refers to vertex u_k and vertices u_{0-i} refers to vertex u_0. Let $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_{s+3}$ be the cops. Then we have the following simple strategy to capture $\mathcal{R}$ using $s + 3$ cops.

The following lemma is central to our strategy.

Lemma 5.4.1. *Let (u_0, u_k) be a dominating pair and let P be a shortest u_0, u_k-path such that $P = u_0, u_1, \ldots, u_k$. Consider a vertex u_i of P. If there is a cop on each of the vertices u_{i-1}, u_i, and u_{i+1} and $\mathcal{R}$ moves to a vertex such that the cop at u_i guards P, then $\mathcal{R}$ will be captured in this cop move.*

Proof. Let the vertex $\mathcal{R}$ moves to be denoted by r. Let $d(u_0, r) = j$. Since P is a dominating path, the vertex r is adjacent to at least one vertex of P, and hence $j \leq k + 1$.

If $d(u_0, r) \leq k$, then $d(u_0, r) = d(u_0, u_i) = i$ since cop at u_i is guarding P. We claim that r can have an edge with a vertex $u \in P$ only if $u \in \{u_{i-1}, u_i, u_{i+1}\}$. Indeed, r cannot have an edge with a vertex $v \in \{u_{i+2}, \ldots, u_k\}$ else $u_0, \ldots, r, v, \ldots, u_k$ becomes a shorter path than P. Similarly, r cannot have an edge with a vertex $v \in \{u_i, \ldots, u_{i-2}\}$ as through the path $u_0, \ldots, v, r$, distance $d(u_0, r)$ becomes less than i. Since $V(P)$ is a dominating set, r must have an edge with at least one of

u_{i-1}, u_i, u_{i+1}. Since all three vertices are occupied by cops, one of the cops will move to capture $\mathcal{R}$.

Similar arguments hold when $d(u_0, r) = k + 1$. In this case, the cops are at vertices u_{k-1} and u_k. Observe that r is only adjacent to vertex u_k in path P. Since there is a cop at u_k, one of the cops will move to capture $\mathcal{R}$.

This completes the proof of this lemma. $\square$

Consider a graph G having a dominating pair u_0, u_k. Let T_i denote the set of vertices such that their graph distance from u_0 is greater than i, that is $T_i = \{x \mid d(u_0, x) > i\}$. We say that $\mathcal{R}$ is *restricted* to T_i, if $\mathcal{R}$ cannot leave the vertices of T_i without getting captured.

Next, we present the main theorem of this section.

Theorem 5.4.1. *Let G be a graph having a dominating pair, then $c_s(G) \leq s + 3$.*

Proof. We will prove this by giving a strategy to capture $\mathcal{R}$ using $s + 3$ cops. Let (u_0, u_k) be a dominating pair in G and let $P = u_0, \ldots, u_k$ be a shortest path u_0, u_k-path. Let the cops be denoted by $\mathcal{C}_1, \ldots, \mathcal{C}_{s+3}$. These cops can capture $\mathcal{R}$ using the following simple strategy. We will prove that this strategy guarantees capture in the rest of the proof.

1. Place cop $\mathcal{C}_i$ on vertex u_{i-1}.

2. In each cop move, if a cop is at vertex $u_i \in \{u_0, \ldots, u_k\}$, then the cop will move to the vertex u_{i+1}.

Observe that when cops place themselves, $\mathcal{R}$ is restricted to T_{s+1}, because, if $\mathcal{R}$ moves to vertex r such that $d(u_0, r) < s + 2$, then it will be guarded by one of the cops from $\mathcal{C}_1, \ldots, \mathcal{C}_{s+2}$ and will be captured immediately (by Lemma 5.4.1).

We claim that when the cop $\mathcal{C}_{s+2}$ moves to vertex u_i, for $i < k$, $\mathcal{R}$ gets restricted to T_i. We prove this by induction on i. For the base case, we showed that when $\mathcal{C}_{s+2}$ is at vertex u_{s+1}, $\mathcal{R}$ is restricted to T_{s+1}. For induction, assume that C_{s+2} is at vertex u_i and R is restricted to T_i. Now, we show that when, in the cop move, C_{s+2} moves to u_{i+1}, $\mathcal{R}$ will be restricted to T_{i+1} after this cop move.

After the cop move, cops are at vertices $u_{i-s}, \ldots, u_{i+2}$ and $\mathcal{C}_{s+2}$ is at vertex u_{i+1}. Let before this cop move, $\mathcal{R}$ was at a vertex r. Since $r \in T_i$, we have $d(u_0, r) > i$. It is sufficient to show that if $\mathcal{R}$ moves to a vertex r' such that $d(u_0, r') \leq i + 1$, then $\mathcal{R}$ will be captured in the next cop move. Let $\mathcal{R}$ moves to such a vertex r'. Since $\mathcal{R}$ has speed s, $d(u_0, r) - s \leq d(u_0, r')$. Hence, $i - s < d(u_0, r') \leq i + 1$, which means one of the cops from $\mathcal{C}_2, \ldots, \mathcal{C}_{s+2}$ guards P and $\mathcal{R}$ will be captured in the next cop move (by Lemma 5.4.1). Thus, $\mathcal{R}$ cannot move to such a vertex and is restricted to T_{i+1}.

Hence, after a finite number of moves, the cop $\mathcal{C}_{s+2}$ is at vertex u_{k-1} and $\mathcal{R}$ is restricted to T_{k-1}. Observe that a vertex $x \in T_{k-1}$, can have an edge only with u_{k-1} or u_k among vertices of P, and have an edge with at least one of them (because vertices of P form a dominating set). Since $\mathcal{R}$ can not move out of T_{k-1} and both u_k and u_{k-1} are occupied by the cops, $\mathcal{R}$ will be captured in the next cop move. $\square$

Since all AT-free graphs have a dominating pair, we have the following immediate corollary.

Corollary 9. *For asteroidal triple free graphs, $c_s \leq s + 3$*

5.5 Classical cops and robber on AT-free graphs

By results of Fomin et al. [46] and our Theorem 5.4.1, the cop number of AT-free graphs in the classical Cops and Robber game is upper bounded by 4. For the *classical* cops and robber game ($s = 1$), we improve this result for AT-free graphs and show that for an AT-free graph G, $c_1(G) \leq 3$ (or $c(G) \leq 3$).

In this section, we consider the classical cops and robber game ($s = 1$), and using a similar algorithm to that in Theorem 5.4.1, we prove that *three* cops are always sufficient to capture a robber in an AT-free graph. Hence we have the following theorem.

Theorem 5.5.1. *Let G be an AT-free graph. Then $c(G) \leq 3$.*

Proof. To prove this, we will give a strategy using three cops to capture the robber in G. Let (u_0, u_k) be a dominating pair of G and let $P = u_0, \ldots, u_k$ be a shortest u_0, u_k-path. Let the three cops be denoted by $\mathcal{C}_1, \mathcal{C}_2$ and $\mathcal{C}_3$. We use the following strategy.

1. Place cop $\mathcal{C}_i$ on vertex u_{i-1}.

2. In each cop move, if a cop $\mathcal{C}$ is at a vertex $u_i \in \{u_0, \ldots, u_k\}$, then will $\mathcal{C}$ will move to the vertex u_{i+1}.

Now, we show that $\mathcal{R}$ will be captured using this strategy in G. Observe that initially $\mathcal{R}$ is restricted to T_1. Now, we will show that, if before a cop move $\mathcal{R}$ is restricted to T_i and in this cop move $\mathcal{C}_2$ moves from u_i to u_{i+1}, then after this cop move $\mathcal{R}$ is restricted to T_{i+1}. Let $\mathcal{R}$ was at a vertex r before this cop move. It is sufficient to show that $\mathcal{R}$ cannot move to a vertex r' such that $d(u, r') \leq i + 1$, without getting captured immediately.

Since $\mathcal{R}$ was restricted to T_i before this cop move, we know that $d(u, r) > i$. Since $\mathcal{R}$ can move at most one edge in a turn, $d(u, r') > i - 1$. Hence, we have to just show that if $i + 1 \geq d(u, r') \geq i$, then $\mathcal{R}$ will be captured immediately. If $d(u, r') = i + 1$, then see that cop $\mathcal{C}_2$ is guarding $\mathcal{R}$ and cops will capture $\mathcal{R}$ using Lemma 5.4.1. If $d(u, r') = i$, then r cannot have an edge with any of u_{i-1}, u_i or u_{i+1}, otherwise one of the cops would have captured $\mathcal{R}$ in last move. Moreover, since $d(u, r') = i$ and $\mathcal{R}$ was restricted to T_i, $d(u, r) = i + 1$. Since P is a dominating path, r is adjacent to at least one vertex of P, hence r is adjacent to u_{i+2}. Now, if r' is adjacent to one of u_i, u_{i+1} or u_{i+2}, then one of the cops will capture $\mathcal{R}$ in the next cop move. For contradiction, let us assume that r' is not adjacent to any of u_i, u_{i+1} or u_{i+2}. Since P is a dominating path, r' has to be adjacent to at least one vertex. Hence, r' is adjacent to u_{i-1}. If this happens, $u_{i-1}, u_i, u_{i+1}, u_{i+2}, r, r', u_{i-1}$ becomes a chordless cycle of length 6 and vertices u_{i-1}, u_{i+1}, r forms an asteroidal triple, which contradicts the fact that G is an AT-free graph. Hence, r' is adjacent to at least one of u_i, u_{i+1} or u_{i+2} and one of the cops will capture $\mathcal{R}$.

This way, when the cop $\mathcal{C}_2$ reaches vertex u_{k-1}, the robber will be restricted to T_{k-1}. Now, a vertex $x \in T_{k-1}$ can have an edge with only u_{k-1} or u_k and have an edge with at least one of them. Since $\mathcal{R}$ cannot move out of T_{k-1} and both u_{k-1} and u_k are occupied by cops, $\mathcal{R}$ will be captured in the next cop move.

This completes the proof of our theorem. □

5.6 Concluding remarks and open problems

In this chapter, we studied the game of cops and robbers on butterfly networks. We showed that the cop number for butterfly networks is 2.

For butterfly networks, in each iteration of the algorithm, a cop guards a level of the network. Conventionally, in the Cops and Robber game on a graph G, a set of cops guard a connected subgraph H of G, and cops stay on the vertices of H. In our strategy, a cop guards a subgraph of the butterfly network that is an independent set, and the cop never enters that subset until it can capture $\mathcal{R}$. We believe that this way of *guarding a disconnected subgraph from a distance* can be useful in finding the cop number of other graph classes.

We gave an asymptotic bound on the capture time of butterfly networks using two cops. It might be interesting to find the exact bounds on the capture time of butterfly networks (assuming optimal play from the robber). Moreover, Luccio and Pagli [79] studied the Cops and Robber game on grids and studied whether increasing the number of cops can decrease the capture time. For a graph, they defined the *work* W_k as $k \cdot capture(k)$, where $capture(k)$ is the number of moves required by k cops to capture the robber. Then they defined *speedup* using $j > i$ cops as W_i/W_j. Since butterfly networks have an inherent structure to support parallel computations, a natural question is whether more cops can work simultaneously to give a speedup greater than one.

The bounds on the cop number for the game of classical Cops and Robber and for the Cops and fast Robber game are not yet tight for AT-free graphs. It would be interesting to further improve these bounds.

6

Conclusion

In this thesis, we studied the game of Cops and Robber and its variants on several graph classes. We studied classical Cops and Robber, Cops and attacking Robber, and lazy Cops and Robber on various kinds of grids. Then we studied the game of Cops and Robber on oriented graphs and studied three models in oriented graphs, namely strong cop model, normal cop model, and weak cop model. Then we considered the game of Cops and Robber on string graphs and boxicity 2 graphs. Finally, we considered some applications of guarding subgraphs to bound the cop number of some graph classes. In this chapter, we give some future directions and some preliminary work related to them.

6.1 Cops, Robber, and Cartesian products

In Chapter 2, we study variants of Cops and Robber on grids. Apart from the subgraphs of grids, the grids considered in this chapter are generated from the Cartesian product of paths and cycles. Let P_m denote a path of length m and C_n denote the cycle on n vertices. We show that for a toroidal grid $(C_m \square C_n)$, one *flexible* and two *lazy* cops can capture the robber. However, the lazy cop number of toroidal grids is still unknown. Hence, we raise the following natural question.

Question 6.1.1. *What is the lazy cop number of toroidal grids?*

We considered the Cartesian product of paths and cycles. It would be interesting to study if these results can be generalized for the Cartesian product of trees and cycles. The classical Cops and Robber game is well studied for the Cartesian product of graphs and tight bounds are known for the cop number of the Cartesian product of trees [80, 83].

Question 6.1.2. *Give bounds on the attacking cop number and lazy cop number of the Cartesian product of trees.*

6.2 Cops and Robber on Oriented graphs

In Chapter 3 we considered the game of Cops and Robber on oriented graphs. We considered three models in oriented graphs depending on the moves allowed to the cops and the robber.

In the open problem session of the GRASTA 2014 [47], N. Nisse asked to characterize the cop-win oriented graphs. In the normal cop model, for a graph $\overrightarrow{G}$ to be cop-win, $\overrightarrow{G}$ must be a directed acyclic graph with a single source. We show that for some graph classes (transitive-triangle free, bipartite, subcubic, outerplanar), a graph $\overrightarrow{G}$ is cop-win if and only if $\overrightarrow{G}$ is a directed acyclic graph with a single source. But the structural characterization of cop-win graphs in the normal cop model remains open. Hence, the following question still remains open.

Question 6.2.1. *Characterize the cop-win graphs for the normal cop model.*

Next, we consider the strong cop model. In the strong cop model, we prove that there for every natural number k, there exists an oriented graph such that the cop number of this oriented graph is more than k. However, the girth and the minimum degree of the graph in our construction are bounded. For the normal cop model, we show that there exist strongly connected oriented graphs with high cop number, high minimum out-degree, and high girth. Hence, the question arises whether such a graph exists for the strong cop model also. Hence, we have the following question.

Question 6.2.2. *For $g, \delta, k \in \mathbf{N}$, does there exists an oriented graph $\overrightarrow{G}$ such that $c_s(\overrightarrow{G}) > k$, girth of $\overrightarrow{G}$ is greater than g, and minimum out-degree of $\overrightarrow{G}$ is greater than δ?*

Also, the open question suggested by N. Nisse [47] to characterize the cop-win graphs in the strong cop model still remains open. Thus, we have the following question.

Question 6.2.3. *Characterize the cop-win graphs in the strong cop model.*

In the weak cop model, we characterize the cop-win graphs.

6.3 Cops and Robber on Intersection graphs

In Chapter 4 we consider the game of classical Cops and Robber on string graphs. Gavenčiak et al. [56] showed that for any string graph G, 15 cops are always sufficient to capture the robber on G, that is, $c(G) \leq 15$. They asked whether these bounds can be improved. They proved that for a shortest path P, 5 cops can guard $N[P]$, and using this proved that the cop number for a string graph is at most 15. We show that if a graph G is *unique shortest path*, then 4 cops can guard $N[P]$. Using this we show that 14 cops are always sufficient to capture the robber in a string graph. Moreover, there is no string graph G known, such that $c(G) > 3$. It would be interesting to study whether these bounds can be further improved. Hence, we have the following question.

Question 6.3.1. *Let* $\mathbf{S}$ *be the class of string graphs. Then we know that* $3 \leq c(\mathbf{S}) \leq 14$. *Can these bounds be improved?*

Beveridge et al. [14] use a similar strategy to show that 9 cops are sufficient to capture the robber in a unit disk graph. It would be interesting to study whether our techniques of handling the unique shortest paths differently can be used there to show that 8 cops are always sufficient to capture the robber in a unit disk graph.

Gavenčiak et al. [56] also showed that for a string graph G with genus g, $c(G) \leq 10g + 15$. Using our techniques, this result can be improved to $c(G) \leq 10g + 14$. It would be interesting to study whether our techniques can be further improved to prove that $c(G) \leq 9g + 14$.

Gavenčiak et al. [56] also proved that for the class of boxicity 2 graphs, 2-Box, $2 \leq c($2-$Box \leq 15$. We improve these bounds and prove that $3 \leq c($2-$Box \leq 14$. However, these bounds are far from tight and it would be interesting to improve these bounds.

6.4 Applications of Guarding subgraphs

In Chapter 5 we study the application of guarding subgraphs of a graph G to bound the cop number of G.

First, we study the game of Cops and Robber on the butterfly networks. Guarding the subgraphs to capture the robber has been used heavily in the Cops and Robber game literature. Conventionally, in the

Cops and Robber game on a graph G, a set of cops guard a connected subgraph H of G, and cops stay on the vertices of H. We use a novel and nontrivial guarding technique where a cop guards a subgraph of the butterfly network that is an independent set, and the cop enters that subgraph only when it can capture $\mathcal{R}$. We believe that this way of *guarding a disconnected subgraph from a distance* can be useful in finding the cop number of other graph classes.

We gave an asymptotic bound on the capture time of butterfly networks using two cops. It might be interesting to find the exact bounds on the capture time of butterfly networks (assuming optimal play from the robber). Moreover, Luccio and Pagli [79] studied the cops and robber game on grids and studied whether increasing the number of cops can decrease the capture time. For a graph, they defined the *work* W_k as $k \cdot capture(k)$, where $capture(k)$ is the number of moves required by k cops to capture the robber. Then they defined *speedup* using $j > i$ cops as W_i/W_j. Since butterfly networks have an inherent structure to support parallel computations, a natural question is whether more cops can work simultaneously to give a speedup greater than one. Hence, we have the following question.

Question 6.4.1. *For an n-dimensional butterfly network, what is the minimum value of work, where work $= k \cdot capture(k)$?*

Next, we consider the game of Cops and fast Robber on the graphs having a dominating pair. Fomin et al. [46] proved that for a graph G having a dominating pair, $c_s(G) \leq 5s - 1$. We improve this bound using a guarding technique and show that $c_s(G) \leq s + 3$. Since AT-free graphs have a dominating pair, this result also holds for AT-free graphs. For the classical Cops and Robber on AT-free graphs, we further improve this bound and show that for an AT-free G, $c(G) \leq 3$. However, these bounds are still not tight and it would be interesting to further tighten these bounds.

6.5 Cops and Robber that can push the graph

Consider the game of Cops and Robber on an oriented graph $\overrightarrow{G}$.

We introduce a new model for oriented graphs where some of the players have the ability to *push* the vertices of the graph. For a vertex v of an oriented graph, the *push* operation on v reverses the orientations of all the arcs incident on v. We define two kinds of push operations that can be performed by players:

- **Weak push:** A player on vertex v having the ability to weak push can either move to an out-neighbour of v or can push v.
- **Strong push:** A player on vertex v having the ability to strong push can either move to an out-neighbour of v or can push *any* vertex of the graph.

Now, a player can have the ability to weak push, strong push, or no ability to push. Depending on what kind of abilities the cops and the robber have, we can have 9 variations of the game, of which the one where neither the cops nor the robber can push is equivalent to the normal cop model.

Observe that if the robber can push, one cop can never capture the robber in an oriented graph $\overrightarrow{G}$ that does not have a vertex u such that every vertex other than u is an out-neighbour of u. Hence, we consider the models where the cop can make a strong/weak push, whereas the robber does not have the push capability.

The main question we motivate is to characterize the cop-win graphs in these 2 models.

LIST OF PUBLICATIONS FROM THE CONTENT OF THIS THESIS

Journal Publications

- S. Das and H. Gahlawat. Variations of cops and robbers game on grids. *Discrete Applied Mathematics*, volume = 305, pages = 340–349, 2021.
- S. Das, H. Gahlawat, U.k. Sahoo, and S. Sen. Cops and Robber on some families of oriented graphs. *Theoretical Computer Science*, volume = 888, pages = 31–40, 2021.

Conference Publications

- S. Das and H. Gahlawat. Variations of cops and robbers game on grids. In *Conference on Algorithms and Discrete Applied Mathematics 2018, CALDAM 2018*, Lecture Notes in Computer Science, volume = 10743, pages = 249–259, Springer, 2018.
- S. Das, H. Gahlawat, U.k. Sahoo, and S. Sen. Cops and Robber on some families of oriented graphs. In *International Workshop on Combinatorial Algorithms 2019, IWOCA 2019*, Lecture Notes in Computer Science, volume = 11638, pages = 188–200, Springer, 2019.
- S.S. Akhtar, S. Das, and H. Gahlawat. Cops and Robber on Butterflies and Solid Grid. In *Conference on Algorithms and Discrete Applied Mathematics 2021, CALDAM 2021*, Lecture Notes in Computer Science, volume = 12601, pages = 272–281, Springer, 2021.

References

[1] I. Adler. Marshals, monotone marshals, and hypertree-width. *Journal of Graph Theory*, 47(4):275–296, 2004.

[2] I. Adler, C. Paul, and D. M. Thilikos. Connected search for a lazy robber. In *39th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2019*, volume 150. Leibniz International Proceedings in Informatics, LIPIcs, 2019.

[3] M. Aigner and M. Fromme. A game of cops and robbers. *Discrete Applied Mathematics*, 8(1):1–12, 1984.

[4] B. Alspach. Sweeping and searching in graphs: a brief survey. *Matematiche*, 59(1-2):5–37, 2004.

[5] A. Amit and N. Linial. Random graph coverings 1: General theory and graph connectivity. *Combinatorica*, 22(1):1–18, 2002.

[6] T. Andreae. On a pursuit game played on graphs for which a minor is excluded. *Journal of Combinatorial Theory Series B*, 41(1):37–47, 1986.

[7] S. Angelopoulos, P. Fraigniaud, F. V. Fomin, N. Nisse, and D. M. Thilikos. Report on GRASTA 2017, 6th Workshop on GRAph

Searching, Theory and Applications. Research report, July 2017. URL: https://hal-lirmm.ccsd.cnrs.fr/lirmm-01645614.

[8] W. Baird and A. Bonato. Meyniel's conjecture on the cop number: A survey. *Journal of Combinatorics*, 3(2):225–238, 2013.

[9] D. Bal, A. Bonato, W. B. Kinnersley, and P. Prałat. Lazy cops and robbers played on random graphs and graphs on surfaces. *Journal of Combinatorics*, 7(4):627–642, 2016.

[10] P. Balister, B. Bollobás, B. Narayanan, and A. Shaw. Catching a fast robber on the grid. *Journal of Combinatorial Theory Series A*, 152:341–352, 2017.

[11] J. Barát. Directed path-width and monotonicity in digraph searching. *Graphs and Combinatorics*, 22(2):161–172, 2006.

[12] A. Berarducci and B. Intrigila. On the cop number of a graph. *Advances in Applied mathematics*, 14(4):389–403, 1993.

[13] D. Berwanger, A. Dawar, P. Hunter, S. Kreutzer, and J. Obdržálek. The dag-width of directed graphs. *Journal of Combinatorial Theory Series B*, 104(4):900–923, 2012.

[14] A. Beveridge, A. Dudek, A. Frieze, and T. Müller. Cops and robbers on geometric graphs. *Combinatorics, Probability and Computing*, 21(6):816–834, 2012.

[15] S. Bhattacharya, G. Paul, and S. Sanyal. A cops and robber game in multidimensional grids. *Discrete Applied Mathematics*, 158(16):1745–1751, 2010.

[16] B. Bollobás, G. Kun, and I. Leader. Cops and robbers in a random graph. *Journal of Combinatorial Theory Series B*, 103(2):226–236, 2013.

[17] A. Bonato, S. Finbow, P. Gordinowicz, A. Haidar, W. B. Kinnersley, D. Mitsche, P. Prałat, and L. Stacho. The robber strikes back. In *proceedings of Computational Intelligence, Cyber Security and Computational Models*, volume 246, pages 3–12. Springer, 2014.

[18] A. Bonato, P. Golovach, G. Hahn, and J. Kratochvíl. The capture time of a graph. *Discrete Mathematics*, 309(18):5588–5595, 2009.

[19] A. Bonato, P. Gordinowicz, W. B. Kinnersley, and P. Prałat. The capture time of the hypercube. *The Electronic Journal of Combinatorics*, 20(2):P24, 2013.

[20] A. Bonato, G. Hahn, and P. Loh. Large classes of infinite k-cop-win graphs. *Journal of Graph Theory*, 65(4):234–242, 2010.

[21] A. Bonato and W. B. Kinnerseley. Bounds on the localization number. *Journal of Graph Theory*, 94(4):579–596, 2020.

[22] A. Bonato and F. Mc Inerney. The game of wall cops and robbers. In *proceedings of Computational Intelligence, Cyber Security and Computational Models*, volume 412, pages 3–13. Springer, 2016.

[23] A. Bonato and B. Mohar. Topological directions in cops and robbers. *Journal of Combinatorics*, 11(1):47–64, 2020.

[24] A. Bonato and R. Nowakowski. *The Game of Cops and Robbers on Graphs*. American Mathematical Society, 2011.

[25] A. Bonato, X. Pérez-Giménez, P. Prałat, and B. Reiniger. The game of overprescribed cops and robbers played on graphs. *Graphs and Combinatorics*, 33:801–815, 2017.

[26] B. Bosek, P. Gordinowicz, J. Grytczuk, N. Nisse, J. Sokół, and M. Śleszyńska Nowak. Localization game on geometric and planar graphs. *Discrete Applied Mathematics*, 251:30–39, 2018.

[27] C. L. Bouton. Nim, a game with a complete mathematical theory. *Annals of Mathematics*, 3(14):35–39, 1901.

[28] N. Bowler, J. Erde, F. Lehner, and M. Pitz. Bounding the cop number of a graph by its genus. *Acta Mathematica Universitatis Comenianae*, 3(12):507–510, 2019.

[29] P. Bradshaw, S. A. Hosseini, and J. Turcotte. Cops and robbers on directed and undirected abelian Cayley graphs. *European Journal of Combinatorics*, In Press, 2021. doi:https://doi.org/10.1016/j.ejc.2021.103383.

[30] S. Brandt, Y. Emek, J. Uitto, and R. Wattenhofer. A tight lower bound for the capture time of the cops and robbers game. *Theoretical Computer Science*, 839:143–163, 2020.

[31] J. Breen, B. Brimkov, J. Carlson, L. Hogben, K. E. Perry, and C. Reinhart. Throttling for the game of cops and robbers on graphs. *Discrete Mathematics*, 341(9):801–815, 2018.

[32] J. Chalopin, V. Chepoi, N. Nisse, and V. Vaxès. Cop and robber games when the robber can hide and ride. *SIAM Journal on Discrete Mathematics*, 25(1):333–359, 2011.

[33] E. Chiniforooshan. A better bound for the cop number of general graphs. *Journal of Graph Theory*, 58(1):45–48, 2008.

[34] N. E. Clarke. *Constrained Cops and Robber*. PhD thesis, Dalhousie University, 2002.

[35] N. E. Clarke, D. Cox, C. Duffy, D. Dyer, S. L. Fitzpatrick, and M. E. Messinger. Limited visibility cops and robber. *Discrete Applied Mathematics*, 282:53–64, 2020.

[36] N. E. Clarke, S. Finbow, and G. MacGillivray. A simple method of computing the catch time. *Ars Mathematica Contemporanea*, 7(2):353–359, 2014.

[37] N. E. Clarke, S. Fiorini, G. Joret, and D. O. Theis. A note on the cops and robber game on graphs embedded in non-orientable surfaces. *Graphs and Combinatorics*, 30(1):119–124, 2014.

[38] N. E. Clarke and G. MacGillivray. Characterizations of k-copwin graphs. *Discrete Mathematics*, 312(8):1421–1425, 2012.

[39] D. G. Corneil, S. Olariu, and L. Stewart. Asteroidal triple-free graphs. *SIAM Journal on Discrete Mathematics*, 10(3):399–430, 1997.

[40] E. Darlington, C. Gibbons, K. Guy, and J. Hauswald. Cops and robbers on oriented graphs. *Rose-Hulman Undergraduate Mathematics Journal*, 17(1):201–209, 2016.

[41] S. G. H. de la Maza, S. A. Hosseini, F. Knox, B. Mohar, and B. Reed. Cops and robbers on oriented toroidal grids. *Theoretical Computer Science*, 857:166–176, 2021.

[42] D. Dereniowski, D. Dyer, R. M. Tifenbach, and B. Yang. The complexity of zero-visibility cops and robber. *Theoretical Computer Science*, 607(2):135–148, 2015.

[43] D. Dereniowski, T. Gavenčiak, and J. Kratochvíl. Cops, a fast robber and defensive domination on interval graphs. *Theoretical Computer Science*, 794:47–58, 2019.

[44] R. Diestel. *Graph Theory.* Springer, 2006.

[45] H. E. Dudeney. *The Canterbury puzzles, puzzle 73.* Dover, 1908.

[46] F. Fomin, P. Golovach, J. Kratochvíl, N. Nisse, and K. Suchan. Pursuing a fast robber on a graph. *Theoretical Computer Science*, 41(7-9):1167–1181, 2010.

[47] F. V. Fomin, P. Fraigniaud, N. Nisse, and D. M. Thilikos. Report on GRASTA 2014, 6th Workshop on GRAph Searching, Theory and Applications. Research report, April 2014. URL: http://www-sop.inria.fr/coati/events/grasta2014/.

[48] F. V. Fomin, P. A. Golovach, and P. Prałat. Cops and robber with constraints. *SIAM Journal on Discrete Mathematics*, 26(2):571–590, 2012.

[49] F. V. Fomin and D. M. Thilikos. An annotated bibliography on guaranteed graph searching. *Theoretical Computer Science*, 399(3):236–245, 2008.

[50] P. Frankl. Cops and robbers in graphs with large girth and Cayley graphs. *Discrete Applied Mathematics*, 17(3):301–305, 1987.

[51] A. Frieze, M. Krivelevich, and P. Loh. Variations on cops and robbers. *Journal of Graph Theory*, 69(4):383–402, 2012.

[52] Z. Gao and B. Yang. The one-cop-moves game on planar graphs. *Journal of Combinatorial Optimization*, 2019. doi:https://doi.org/10.1007/s10878-019-00417-x.

[53] M. Gardner. Mathematical games - the fantastic combinations of John Conway's new solitaire game "life". *Scientific American*, 223:120–123, 1970.

[54] M. Gardner. Mathematical games - cram, crosscram and quadraphage: new games having elusive winning strategies. *Scientific American*, 230(2):106–108, 1974.

[55] T. Gavenčiak. Cop-win graphs with maximum capture-time. *Discrete Mathematics*, 310(10-11):1557–1563, 2010.

[56] T. Gavenčiak, P. Gordinowicz, V. Jelínek, P. Klavík, and J. Kratochvíl. Cops and robbers on intersection graphs. *European Journal of Combinatorics*, 72:45–69, 2018.

[57] A. C. Giannopoulou, P. Hunter, and D. M. Thilikos. Lifo-search: A min–max theorem and a searching game for cycle-rank and tree-depth. *Discrete Applied Mathematics*, 160(15):2089–2097, 2012.

[58] A. S. Goldstein and E. M. Reingold. The complexity of pursuit on a graph. *Theoretical Computer Science*, 143(1):93–112, 1995.

[59] P. A. Golovach. Equivalence of two formalizations of a search problem on a graph. *Vestnik Leningrad. Univ. Mat. Mekh. Astronom.*, pages 10–14, 1989.

[60] P. A. Golovach. A topological invariant in pursuit problems. *Differ. Uravn.*, 25:923–929, 1989.

[61] G. Hahn and G. MacGillivray. A note on k-cop, l-robber games on graphs. *Discrete Mathematics*, 306(19-20):2492–2497, 2006.

[62] Y. O. Hamidoune. On a pursuit game on Cayley digraphs. *European Journal of Combinatorics*, 8(3):289–295, 1987.

[63] S. A. Hosseini. *Game of Cops and Robbers on Eulerian Digraphs.* PhD thesis, Simon Fraser University, 2018.

[64] S. A. Hosseini and B. Mohar. Game of cops and robbers in oriented quotients of the integer grid. *Discrete Mathematics*, 341(2):439–450, 2018.

[65] P. Hunter and S. Kreutzer. Digraph measures: Kelly decompositions, games, and orderings. *Theoretical Computer Science*, 399(3):206–219, 2008.

[66] W. Imrich and S. Klavžar. *Product graphs, Structure and Recognition.* John Wiley & Sons Inc., 2000.

[67] V. Isler, S. Kannan, and S. Khanna. Randomized pursuit-evasion with local visibility. *SIAM Journal on Discrete Mathematics*, 20(1):26–41, 2006.

[68] T. Johnson, N. Robertson, P. D. Seymour, and R. Thomas. Directed tree-width. *Journal of Combinatorial Theory Series B*, 82(1):138–154, 2001.

[69] G. Joret, M. Kaminski, and D. O. Theis. Cops and robber game on graphs with forbidden (induced) subgraphs. *Contributions to Discrete Mathematics*, 5(2):40–51, 2010.

[70] A. Kehagias. Generalized cops and robbers: A multi-player pursuit game on graphs. *Dynamic Games and Applications*, 9:1076–1099, 2019.

[71] D. Khatri, N. Komarov, A. Krim-Yee, N. Kumar, B. Seamone, V. Virgile, and A. Xu. A study of cops and robbers in oriented graphs. *arXiv:1811.06155*, 2019.

[72] W. B. Kinnersley. Cops and robbers is exptime-complete. *Journal of Combinatorial Theory Series B*, 111:201–220, 2015.

[73] W. B. Kinnersley. Bounds on the length of a game of cops and robbers. *Discrete Mathematics*, 341(9):2508–2518, 2018.

[74] G. Konstantinidis and A. Kehagias. Simultaneously moving cops and robbers. *Theoretical Computer Science*, 645:48–59, 2016.

[75] F. Lehner. On the cop number of toroidal graphs. *arXiv:1904.07946*, 2020.

[76] F. Leighton. *Introduction to parallel algorithms and architectures.* Morgan Kaufmann, 1992.

[77] P. Loh and S. Oh. Cops and robbers on planar directed graphs. *Journal of Graph Theory*, 86(3):329–340, 2017.

[78] L. Lu and X. Peng. On Meyniel's conjecture of the cop number. *Journal of Graph Theory*, 71(2):192–205, 2011.

[79] F. Luccio and L. Pagli. Cops and robber on grids and tori: basic algorithms and their extension to a large number of cops. *The Journal of Supercomputing*, 2021. doi:https://doi.org/10.1007/s11227-021-03655-1.

[80] M. Maamoun and H. Meyniel. On a game of policemen and robber. *Discrete Applied Mathematics*, 17(3):307–309, 1987.

[81] M. Mamino. On the computational complexity of a game of cops and robbers. *Theoretical Computer Science*, 477:48–56, 2013.

[82] A. Mehrabian. The capture time of grids. *Discrete Mathematics*, 311(1):102–105, 2011.

[83] S. Neufeld and R. Nowakowski. A game of cops and robbers played on products of graphs. *Discrete Mathematics*, 186(1-3):253–268, 1998.

[84] N. Nisse. Network decontamination. In *Distributed Computing by Mobile Entities, Current Research in Moving and Computing*, volume 11340 of *Lecture Notes in Computer Science*, pages 516–548. Springer, 2019.

[85] N. Nisse and K. Suchan. Fast robber in planar graphs. In *Graph-Theoretic Concepts in Computer Science. WG 2008*, volume 5344. Lecture Notes in Computer Science, Springer, 2008.

[86] R. Nowakowski and P. Winkler. Vertex-to-vertex pursuit in a graph. *Discrete Mathematics*, 43(2-3):235–239, 1983.

[87] R. J. Nowakowski and R. S. W. Schröder. Bounding the cop number using the crosscap number. *preprint*, 1997.

[88] D. Offner and K. Ojakian. Variations of cops and robber on the hypercube. *Australasian Journal of Combinatorics*, 59(2):229–250, 2014.

[89] T. D. Parsons. Pursuit-evasion in a graph. In *Theory and Applications of Graphs, LNCS*, volume 642, pages 426–441. Springer, 1978.

[90] T. D. Parsons. The search number of a connected graph. In *Proceedings of the Ninth Southeastern Conference on Combinatorics, GraphTheory, and Computing*, volume XXI, pages 549–554. Utilitas Math., 1978.

[91] N. N. Petrov. A problem of pursuit in the absence of information on the pursued. *Differ. Uravn.*, 18:1345–1352, 1982.

[92] P. Pisantechakool and X. Tan. The capture time of a planar graph. *The Electronic Journal of Combinatorics*, 36(4):1103–1117, 2018.

[93] P. Prałat and N. Wormald. Meyniel's conjecture holds for random graphs. *Random Structures and Algorithms*, 48(2):396–421, 2015.

[94] P. Prałat and N. Wormald. Meyniel's conjecture holds for random d-regular graphs. *Random Structures and Algorithms*, 55(3):719–741, 2019.

[95] A. Quilliot. *Thése d'Etat*. PhD thesis, Université de Paris VI, 1983.

[96] A. Quilliot. A short note about pursuit games played on a graph with a given genus. *Journal of Combinatorial Theory Series B*, 38(1):89–92, 1985.

[97] B. S. W. Schroeder. The copnumber of a graph is bounded by $\lfloor 3/2 * genus(g) \rfloor + 3$. In *Categorical perspectives. Trends in mathematics*, pages 243–263, 2001.

[98] A. Scott and B. Sudakov. A bound for the cops and robbers problem. *SIAM Journal on Discrete Mathematics*, 25(3):1438–1442, 2011.

[99] P. D. Seymour and R. Thomas. Graph searching and a min-max theorem for tree-width. *Journal of Combinatorial Theory Series B*, 58(1):22–33, 1993.

[100] Y. Shitov. Two remarks on the game of cops and robbers. *Bulletin of the Korean Mathematical Society*, 47(1):127–131, 2020.

[101] K. A. Sim, T. S. Tan, and K. B. Wong. Lazy cops and robbers on generalized hypercubes. *Discrete Mathematics*, 340(7):1693–1704, 2017.

[102] R. Stahl. *Computability Theoretic Results for the Game of Cops and Robbers on Infinite Graphs*. PhD thesis, University of Connecticut, 2017.

[103] D. O. Theis. The cops and robber game on series-parallel graphs. *arxiv:0712.2908*, 2008.

[104] R. Tošić. Vertex-to-vertex search in a graph. In *proceedings of the Sixth Yugoslav Seminar on Graph Theory*, pages 233–237, 1985.

[105] E. J. van Leeuwen. A cover-based approach to multi-agent moving target pursuit. In *proceedings of the Fourth Artificial Intelligence and Interactive Digital Entertainment Conference*, pages 54–59, 2008.

[106] L. Wang and B. Yang. The one-cop-moves game on graphs with some special structures. *Theoretical Computer Science*, 847:17–26, 2020.

[107] D. B. West. *Introduction to Graph Theory*. Prentice Hall, 2000.

[108] K. Zarankiewicz. Problem P101. *Colloquium Mathematicum*, 3:19–30, 1954.