10-28-2024

The Graph Burning and The Firefighter Problems

Jilsa Chandarana
University of Windsor

The Graph Burning and The Firefighter Problems

By

Jilsa Chandarana

A Thesis
Submitted to the Faculty of Graduate Studies
through the School of Computer Science
in Partial Fulfillment of the Requirements for
the Degree of Master of Science
at the University of Windsor

Windsor, Ontario, Canada

2024

The Graph Burning and The Firefighter Problems

by

Jilsa Chandarana

APPROVED BY:

D. Borisov
Department of Mathematics and Statistics

C. Bright
School of Computer Science

A. Biniaz, Advisor
School of Computer Science

October 11, 2024

DECLARATION OF ORIGINALITY

I hereby certify that I am the sole author of this thesis and that no part of this thesis has been published or submitted for publication.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis and have included copies of such copyright clearances to my appendix.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

ABSTRACT

With the increasing usage of social media, the spread of information has become abundant. However, this surge in accessible news and updates has also heightened the risk of spreading rumors. In our research, we explore two problems related to this phenomenon: the graph burning problem and the firefighter problem.

We provide a comprehensive survey of the graph burning problem, a discrete-time process. Initially, all the vertices of the graph are unburned, and we start the fire at a single vertex. The fire then spreads to the adjacent vertices of the burned vertices, and at each step, we select a new vertex to burn. Our objective is to burn all the vertices in a minimum number of steps. The graph burning problem is NP-Complete. We review various approximation algorithms and bounds, highlighting significant advancements over the past decade.

In the firefighter problem, a graph is given, and a fire starts from a subset of vertices. At each step, firefighters protect a certain number of vertices as the fire continues to spread to adjacent vertices.

DEDICATION

To my friends and fellow classmates, whose companionship and shared experiences have been invaluable throughout this journey. Your support, laughter, and motivation have helped me push through challenges, and I'm grateful to have you all by my side.

ACKNOWLEDGEMENTS

I would like to express my heartfelt and deepest gratitude to my supervisor Dr. Biniaz, for his invaluable support and encouragement during this journey. I would also like to express my sincerest appreciation to the insightful members of my thesis committee, including Dr. Borisov and Dr. Bright.

I would also like to thank Arash for useful discussions and suggestions and Mustafa, who participated in meetings. Your presence made it better.

I am deeply grateful to my friends, colleagues, and classmates because the support and friendship they have shown have made a huge difference in making this academic journey easier.

Lastly, I want to express deep appreciation to my family for continued support in every step of my life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

Introduction

Communication has always played an important role in human life throughout history. Over time, humans have developed increasingly sophisticated methods to communicate over long distances.

In the modern era, the Internet has become a part of everyday life for everyone. The Internet allows us to share information instantly with any user anywhere in the world. This development has encouraged us to study the spread of the information [4]. However, it is not only useful information that gets spread over a network. Sometimes it is falsified information or rumors that we do not want to spread. It can also be compared with the spread of a virus or epidemic through a network [13]. Both the problems have been widely studied as the mathematical models of graph burning [4] and the firefighter problem [17].

In our thesis, we are going to discuss the important research milestones in these fields. In section 1, we discuss the general introduction for the problems while section 1.1 focuses on graph burning and section 1.2 focuses on the firefighter problem. In section 1.3, we have introduced all the useful terminology and concepts that are used throughout the thesis. Section 1.4 describes the background where the motivation and usefulness of these problems are noted while section 1.5 discusses the current best results that have been studied over the years.

Section 2 focuses on the Euclidean version of the graph burning. It heavily depends on the minimum dominating set [16, 19, 26] which is discussed in section 2.1. Since the problem is NP-Complete [19], many approximation algorithms have been developed [16, 19, 26] and some of them are discussed in section 2.2. Subsections 2.2.1 to 2.2.3 study these algorithms and provide a sketch of the proof which shows the incremental advancement of the research from $(2 + \epsilon)$-Approximation [19] in subsection 2.2.1 to $(1.96296 + \epsilon)$-Approximation [16] in subsection 2.2.2 to $(1.944 + \epsilon)$-Approximation [26] in subsection 2.2.3.

Section 3 discusses the Graph Burning problem in detail. Many of the useful properties have been proposed [4] which are presented in section 3.1. This problem is also proven to be NP-Complete [2]. Section 3.2 discusses the approximation algorithms proposed. For general graphs, Subsection 3.2.1 shows a 3-Approximation algorithm [5] and subsection 3.2.2 shows a randomized 2.313-Approximation algorithm [21]. For trees, a 2-Approximation algorithm [5] has been proposed in subsection 3.2.3. Subsection 3.2.4 discuss the algorithm based on a greedy approach which gives a $(3 - \frac{2}{b(G)})$-approximation [14]. At last, the approximation algorithm for directed graphs has been studied in subsection 3.2.5 where the algorithm for multi-rooted trees [15] is discussed in 3.2.5.1 and the algorithm for single-rooted trees [15] is discussed in 3.2.5.2. Various combinatorial bounds have also been proposed for graph burning which is discussed in section 3.3 [1, 3, 4]. Depending on the graph structures, the sections are divided. Bounds for general graphs are given in subsection 3.3.1 [1, 3, 4] where we propose our finding related to the current bound in subsection 3.3.1.1, for trees in subsection 3.3.2 [10], for graphs with minimum degree k in section 3.3.3 [20], for directed graphs in subsection 3.3.4 [18] and for spider graphs in subsection 3.3.5 [9].

The firefighter problem is comprehended in section 4. The previous results are summarised in section 4.1 [7, 11, 22].

Section 5 shows the summary and conclusion of the thesis is given in section 5.1 along with the future work in section 5.2.

1.1 Graph Burning

The rise of social media has resulted in an efficient and swift exchange of information, which gives us ease to spread information or news. The information travels from one user to another. This phenomenon was mathematically modelled as the Graph Burning problem by Anthony Bonato in 2014 [4]. In this model, the network is analogous to the graph structure where the nodes denote the social media users and the edges represent the connections between them. Burning a node means giving a piece of news to that user.

There are several variants of the graph burning problem, which can be broadly classified into classic versions and Euclidean versions. The classic graph version, as previously discussed, involves nodes and edges where the fire spreads to the neighboring nodes. In contrast, the Euclidean version does not involve a graph structure but instead consists of a set of points in the Euclidean plane. In this variant, one point is burned at each step, and in the subsequent step, all points within a 1-unit radius from a burned point are also burned, while another point is chosen to start burning. Once a vertex or point is burned, it stays in that state for the rest of the process. The process ends when all the nodes or given points are burned. It is hard to decide which points to burn and even if we know those points, it is hard to decide the order in which we should burn them [2].

1.1.1 Classic Version

In the context of social media, we can relate the graph burning model by considering all users as vertices (nodes) in a graph, with the edges representing the connections between them [4]. The burning process is a discrete-time process that shows the spread of information across the network. Initially, one vertex (a user) receives the news or information. It means we have burned that node. In the next time step, the information is spread by that user to his connections which means the burned node burns its neighbors. At the same time, a new user independently gets the news which acts as the next burning source. In each step, the fire (news) keeps spreading to the neighbors while new sources are picked. The process continues until all nodes in the graph (all the users in the network) are burned (get the news). This model depicts the characteristic spread of information in the network which can be studied mathematically as a new graph parameter [4].

1.1.2 Euclidean Version

The Euclidean graph burning problem was inspired by the classic graph burning version [19]. The aim of both the variants remains same which is to burn all the points in the minimum number of steps but since the fundamental structure changes, it drastically affects the burning process.

In the classical graph burning model, a vertex is burned at each step, and the fire spreads to its neighboring vertices in the next step, and so on. In contrast, Euclidean graph burning is a process of burning a set of points in the 2-dimensional Euclidean plane. In this version, the process begins by choosing one of the points as a burning source. In the next step, the fire spreads to all points within a radius of 1 unit. Any points within that distance from the burning source are immediately burned. Additionally, a new burning source is chosen out of the remaining unburned points. The fire continues to spread until all the points are burned.

The main challenge of the problem lies in selecting the points to burn the remaining points in the fewest steps possible. As the fire spreads radially outwards in each step, we can think of it as covering the points with the disks of varying radii [16, 19, 26].

1.2 Firefighter Problem

The graph burning problem is derived from the firefighter problem [4]. In the firefighter problem, along with the graph, a set of burning sources is given. Alongside, we have a certain number of firefighters and the aim is to protect as many vertices as possible from the fire [13]. At each step, the fire spreads similarly to the graph burning process and the firefighters protect a set of vertices. Once a vertex is protected by a firefighter, the fire can no longer spread through it. The process ends when the fire is no longer able to spread to any new vertex. When a vertex is burned or protected, it stays in that state for the rest of the process. The main challenge in this problem is to find the optimal vertices to deploy firefighters so that we can protect maximum number of vertices [13].

The firefighter problem can be thought of as a turn-based game [7]. Initially, the fire starts at the predefined sources and we deploy the given number of firefighters. In the next step, the fire spreads to the unprotected neighbors of the burned vertices and then the firefighters proceed to protect a new set of unburned vertices. In the original firefighter problem, the firefighters were not restricted in terms of movements [13] but in 2022, Burgess et al. proposed a constrained version of the firefighter problem where the firefighters are allowed to move at most certain distance [7]. This modification resembles a real-life scenario and has different results than the original version that needs to be studied [7].

1.3 Basic Terminology and Concepts

Before proceeding further, we will define some key terms and concepts that will be referenced throughout the rest of the thesis.

- **Burning a Graph:** Burning is a discrete-time process defined on a graph

$G(V, E)$. Any vertex of a graph is either burned or unburned. Initially, all vertices are unburned. The burned vertices stay burned until the end of the process. At $t = 1$, we choose a vertex to burn. For $t = 1, 2, 3, \ldots$, the burned vertices burn their neighbour vertices and we pick one extra vertex to burn. The process ends when all the vertices of the graph are burned.

- **Burning a set of points:** In the Euclidean version of the burning process, a set of points S is given in $\mathbb{R}^2$. At each discrete time step, one point is selected and burned. After this initial burning, in the next step, all points within a specified radius (typically 1 unit) of the burned point are also burned. This process continues with a new point being burned in each subsequent step and the expansion of fire from previous burning sources. The process ends when all points are burned.

- **Burning Number:** The minimum number of steps required to burn the graph is called the burning number of that graph. It is denoted by $b(G)$ where G is an input graph.

- **Burning Sequence:** The order in which the vertices are chosen to burn is called the burning sequence.

- **Burning Sources:** A vertex in the burning sequence is known as a burning source.

For example, consider a 7×7 square grid, and call it G, as shown in the figure 1.3.1. We want to burn it in a minimum number of steps.

Steps to burn 7×7 square grid:

1. The process begins by burning the middle vertex, which is highlighted in red.

2. In the next step, the fire spreads to the vertices adjacent to the middle vertex, shown in grey. At the same time, we select a new burning source in the top-left grid.

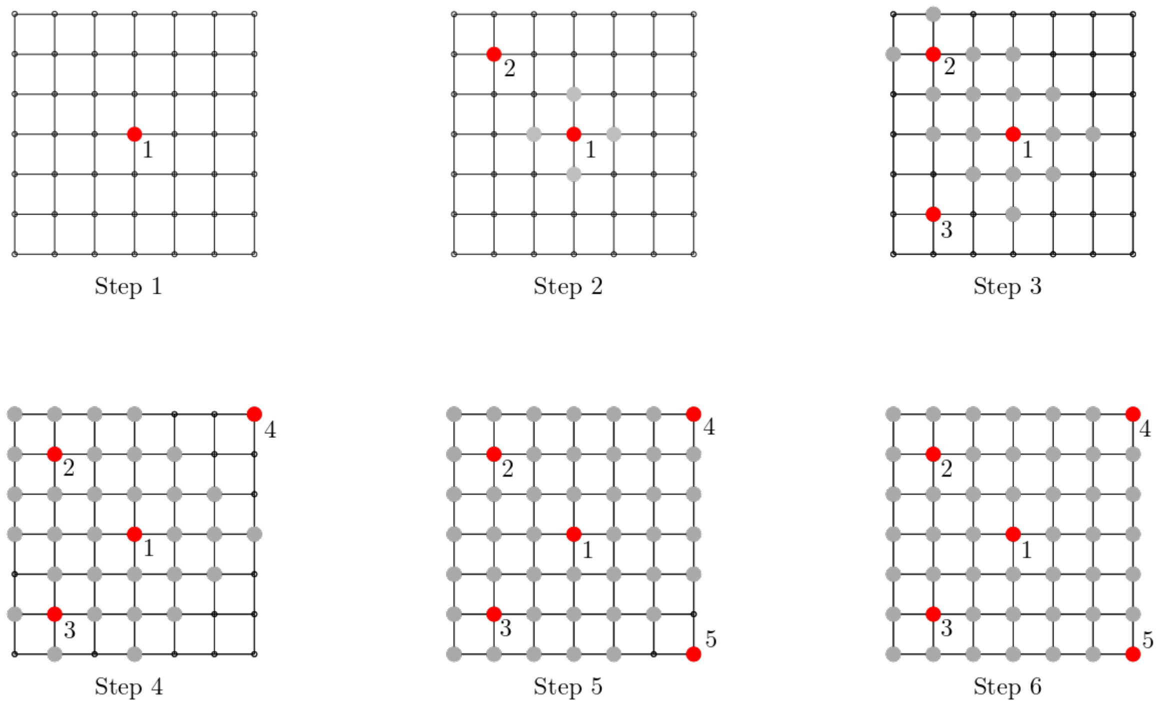

Fig. 1.3.1: Burning a 7×7 square grid

3. The fire continues to spread from both the initial burning source and the newly chosen one. The adjacent vertices of both burning sources are burned, and we select another burning source in the bottom side of the grid.

4. In this step, the fire keeps spreading from all active burning sources, and an additional burning source is selected at the top-right corner of the grid.

5. The burning process continues, with the fire expanding from all current burning sources. We place an additional burning source at the bottom-right corner.

6. By this point, the fire has spread to all vertices, and the process ends. Since all vertices are already burned, we do not need to choose the burning source in this step.

Thus, using this method, we conclude that the burning number of the graph, $b(G)$, is at most 6.

The firefighter problem differs significantly from the graph burning process.

- **Firefighting Process:** Firefighting is a discrete-time process. In a graph $G(V, E)$, a set $S \subset V$ of burning sources is given. With b firefighters, we can protect up to b vertices in each step. After the firefighters are deployed, the fire spreads to any adjacent, unprotected vertices and the firefighters protect the new set of vertices. The process continues until the fire can no longer spread, either because all vertices are protected or there are no more unprotected neighbors to ignite [13].

For example, let us consider the graph given in figure 1.3.2. In this example, the fire starts at the top vertex, which acts as the source of the fire and we have only one firefighter.

Steps for Firefighting:

1. The firefighter is placed on the neighboring vertex u_1, stopping the fire from spreading to this vertex.

2. After the firefighter's action, it's the fire's turn to spread. Since u_1 is protected, the fire moves to the next unprotected vertex to the right.

3. In this step, the firefighter protects vertex u_2, preventing the fire from spreading to this point.

4. The fire spreads again, moving to another unprotected neighboring vertex. The firefighter can now protect u_3.

Once the firefighter has protected u_3, the fire is contained. Since there are no unprotected neighboring vertices left for the fire to spread to, the process ends.

1.4 Background

1.4.1 Graph Burning

Graph burning is a theoretical model designed to analyze the spread of information, such as news or rumors, across a network. The main challenge is to decide which ver-

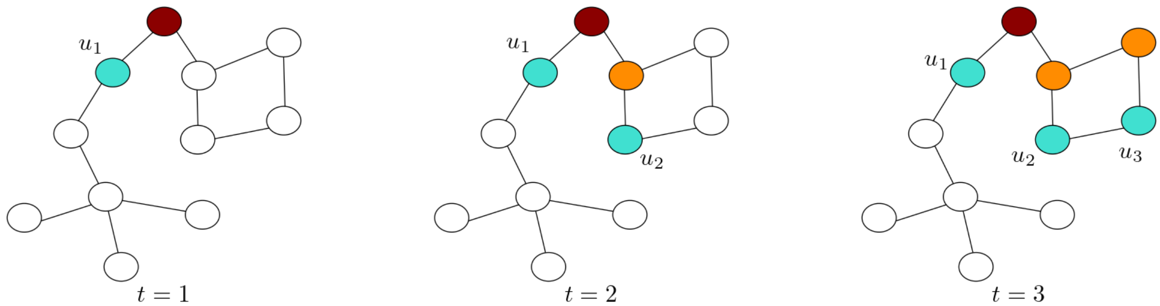

Fig. 1.3.2: Process of Firefighter on a graph

tices or points to burn in each step to complete the burning process in the minimum number of steps. Even when we know which vertices to burn, it is difficult to decide their order of burning. Because of that graph burning problem is NP-Complete [2]. Bonato et al proved that the graph burning problem is reducible from a distinct 3-partition problem which is a famous NP-Complete problem [2].

Over the years, various studies have been conducted to establish upper bounds on the steps required to burn an entire graph [1, 3, 4]. Researchers have proposed a number of approximation algorithms to find near-optimal solutions in feasible time [5, 21]. Yet, even at present, much research is in progress to get better bounds and better approximation ratios.

The graph burning problem has been studied for a variety of graph types, resulting in different bounds and characteristics unique to them [5, 9, 10, 23]. For example, many papers have been published that focused on the burning process for trees [5, 10], disjoint paths [5], spider graphs [9] and so on.

The graph burning process is also studied for directed graphs [15, 18]. The burning process differs for the directed graphs because in this case, the fire only spreads to the outgoing neighbors [18].

1.4.1.1 Euclidean Version

The Euclidean version of the graph burning problem was proposed by Keil et al. in 2022 [19]. It extends the concept of graph burning to a set of points in the Euclidean plane. Instead of vertices and edges, it deals with points distributed in a 2-dimensional Euclidean plane. Same as in the classic version, the aim is to burn the points in the minimum number of steps [19]. In this model, at each discrete time step, one point is burned, and subsequently, the fire spreads from burned points to additional 1 unit radius.

As discussed before, the main challenge is to find which points to burn in each step to minimize the total time required to burn all points in the plane. Keil et al. demonstrated that this problem is NP-hard [19]. In the Euclidean version of graph burning, two main variants have been studied: point burning and anywhere burning [19].

- **Point Burning:** The burning process must start from one of the given points in the plane. At each step, a point from the predefined set is selected as the burning source. The fire then spreads to all points within a specified radius.
- **Anywhere Burning:** The anywhere burning variant allows the burning to begin from any location in the plane, not necessarily at one of the predefined points.

Figure 1.4.1 illustrates both the two variants of Euclidean graph burning.

The major difference between both versions is the placement of the burning source [19]. It is obvious that the flexibility of anywhere burning can result in better placement of the burning source that can potentially reduce the overall steps required to burn the given points.

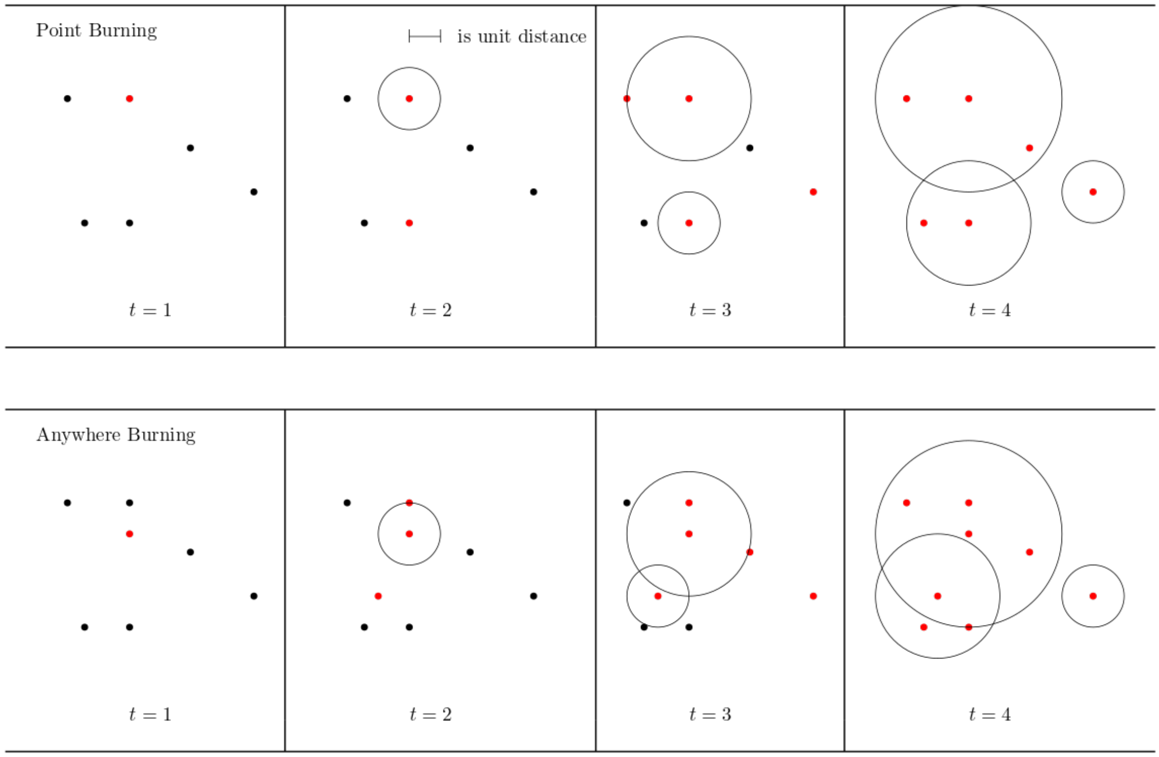

Fig. 1.4.1: Burning Points in the Plane

1.4.2 Firefighter Problem

The firefighter problem was introduced by Bert Hartnell in 1995 [17]. As Graph burning deals with the spread of information, the firefighter problem works to limit the spread. The classic firefighter problem traditionally deals with connected graphs, where the goal is to strategically deploy firefighters to protect vertices and prevent the spread of fire. The problem was proven to be NP-hard [12]. The complexity of the problem lies in the strategy to find optimal vertices to deploy the firefighters. Over the time, various algorithms and heuristics have been developed to deal with this problem which are surveyed in the literature. [13].

In 2022, Burgess et al. [7] introduced a new variant of the firefighter problem known as the distance-restricted firefighter problem.

- **Distance-Restricted Firefighting:** In the distance-restricted firefighting process, firefighters are constrained to move a maximum of a certain distance. Specifically, in a graph $G(V, E)$, firefighters can only move to vertices that are within a distance of d, where the distance between two vertices is defined as the number of edges in the shortest path connecting them. In each step, with b firefighters available, up to b vertices can be protected, given that the distance between the firefighter's current position and the next position is at most d. The fire spreads to unprotected neighboring vertices of the burned vertices, and the process continues until the fire can no longer spread [7].

The distance-restricted version of firefighting resembles the real-life scenario because, in practical life, firefighters can only move a certain distance in a given time [7].

1.5 Related Works

In this section, we will explore various research advancements in the fields of graph burning and the firefighter problem. Both areas have received significant attention in the past few years due to their practical applications. We already know that researchers have proposed various approximation algorithms [5, 21] along with the mathematical bounds [1, 3, 4] that have given us better insights into these fields.

1.5.1 Graph Burning

As previously mentioned, the graph burning problem was introduced by Anthony Bonato in 2014 [4]. In his foundational work, he established several important properties of the problem, along with the very famous graph burning conjecture.

Conjecture 1. *(Bonato, 2014) For any graph with n vertices, $b(G) \leq \lceil \sqrt{n} \rceil$.*

The burning conjecture is still not proven. Bonato et al. [4] also established an upper bound for general graphs. They demonstrated that any graph with n vertices can be burned in at most $2\sqrt{n} - 1$ steps [4].

In 2017, Bessi et al. [2] proved the NP-completeness of the graph burning problem with a reduction from distinct-3-partitions. In 2018, the upper bound was improved by Bessi et al. to $\sqrt{\frac{12n}{7}}$ [3]. Also in 2018, Das et al. proved that the burning conjecture holds for spider graphs [9]. The spider graphs are a special class of graphs where there is one vertex of degree at least 3, and the rest of the vertices have degree less than 3.

In 2019, Bonato and Kamali [5] proposed a 3-approximation algorithm to burn general graphs and a 2-approximation algorithm to burn trees. In 2020, Janssen [18] extended graph burning to directed graphs, where fire can only spread to outgoing neighbors, and proved the problem of burning directed graphs is NP-hard.

In 2021, Bastide et al. [1] further improved the upper bound of the general undirected graph to $\sqrt{\frac{4n}{3}}$, which is currently the best known bound. In 2022, Garcia-Diaz et al. [14] proposed the farthest-first algorithm based on a greedy approach, where the next burning source is chosen at the farthest distance from the previously chosen sources. This resulted in a $3 - \frac{2}{b(G)}$ approximation, where $b(G)$ is the optimal burning number of the input graph G [14].

In 2023, Martinsson [21] proposed a randomized algorithm for graph burning, achieving a 2.313-approximation algorithm. In the same year, Gautam et al. [15] developed a 3-approximation algorithm to burn multi-rooted directed trees and introduced a modified version for single-rooted trees with a 1.905-approximation.

Recently, in 2024, Norin and Turcotte [25] proved the burning conjecture asymptotically, meaning they demonstrated that for any graph G of n vertices, $b(G) \leq (1 + o(1))\sqrt{n}$ where $o(1)$ means a function of n that approaches 0 as n approaches infinity.

1.5.1.1 Euclidean Version

The Euclidean version of the graph burning problem was introduced by Keil et al. in 2022 [19]. They proved the problem to be NP-complete and also proposed a $(2+\epsilon)$-approximation algorithm for it [19]. In 2023, Gokhale et al. [16] proposed an improvement which resulted in a $(1.96296+\epsilon)$-approximation algorithm.

In 2024, Kamali and Shabanijou [26] developed a $(1.944+\epsilon)$-approximation algorithm, which currently stands as the best-known approximation for the point burning problem.

1.5.2 Firefighter Problem

As discussed before, the firefighter problem was introduced by Hartnell in 1995 [17]. In 2007, Finbow et al. [12] proved that the firefighter problem is NP-complete, even the decision version of problem for trees with maximum degree 3 by reducing it from the Not-All-Equal 3SAT problem. There is a vast number of publications dedicated to researching the firefighter problem, reflecting the complexity and practical applications. A comprehensive survey of the firefighter problem is provided by Finbow and MacGillivray [13]. In this work, we are going to focus specifically on the distance-restricted firefighting version.

In 2005, Messinger [22] proposed a thesis on firefighting in the infinite grid. They discuss about both grids: square grid and strong grid. The strong grid is where a vertex is connected with eight of its neighbors. They proved that four firefighters are necessary to contain the fire in the square grid, even without any distance restriction [22]. In 2019, Days-Merrill [11] determined the number of required firefighters for $d = 1$. They proved that for the square grid, we need at least four firefighters, while in the strong grid, at least eight firefighters are required to contain the fire when firefighters are restricted to move at most one unit in each step [11]. In 2022, Burgess et al. [7] calculated that when firefighters can move at most two units ($d = 2$), at least four firefighters are needed for the strong grid, and three firefighters are sufficient to contain the fire in the square grid. They also conjectured that two firefighters are not enough to contain the fire in the square grid when $d = 2$.

Conjecture 2. *(Burgess et al., 2022) On the infinite square grid, two firefighters are not sufficient to contain the fire.*

The table 1.5.1 presents the known results for both the square and strong grids.

Type of Grid	$d = 1$	$d = 2$	$d \geq 3$
Strong Grid	8 [11]	4 [7]	4 [22]
Square Grid	4 [11]	2 or 3 [1] [7]	2 [7]

Table 1.5.1: Necessary number of firefighters to contain fire in infinite grid

In 2023, Burgess et al. [6] proposed the distance-restricted firefighting problem for finite graphs and proved that it is NP-Hard.

[1] Conjecture 2 says that we can not contain the fire in a square grid with 2 firefighters when $d = 2$ but since the conjecture is not proven yet, we are not sure about the minimum required number of firefighters.

CHAPTER 2

Graph Burning - Euclidean Version

After almost a decade of the introduction of graph burning, Keil et al. [19] proposed a new variant. This variant proposed the burning number for the points in the 2-dimensional Euclidean plane $\mathbb{R}^2$. Unlike the classic version, there is no graph with vertices or edges. That changes the definition of the burning process drastically.

As mentioned before, the are two types of Euclidean Graph Burning: Point Burning and Anywhere Burning. In this chapter, we are going to focus only on the point-burning version.

The burning process for the points in the plane can be explained this way [19]. We have a set of points in the Euclidean plane. The burning process starts by selecting one of the given points as a burning source and calling it v_1. In the next step, all the points within 1 unit radius of v_1 will get burned. After that, we pick another point out of the remaining unburned points as the new burning source at this time and call it v_2. In the following step, the fire from v_1 will be spread to 2-unit radius while v_2 will burn every point within a 1-unit radius and then we pick v_3. The fire keeps spreading and the process ends when all the points are burned [19].

For a better understanding of the point burning process, let us consider the example given in Figure 2.0.1.

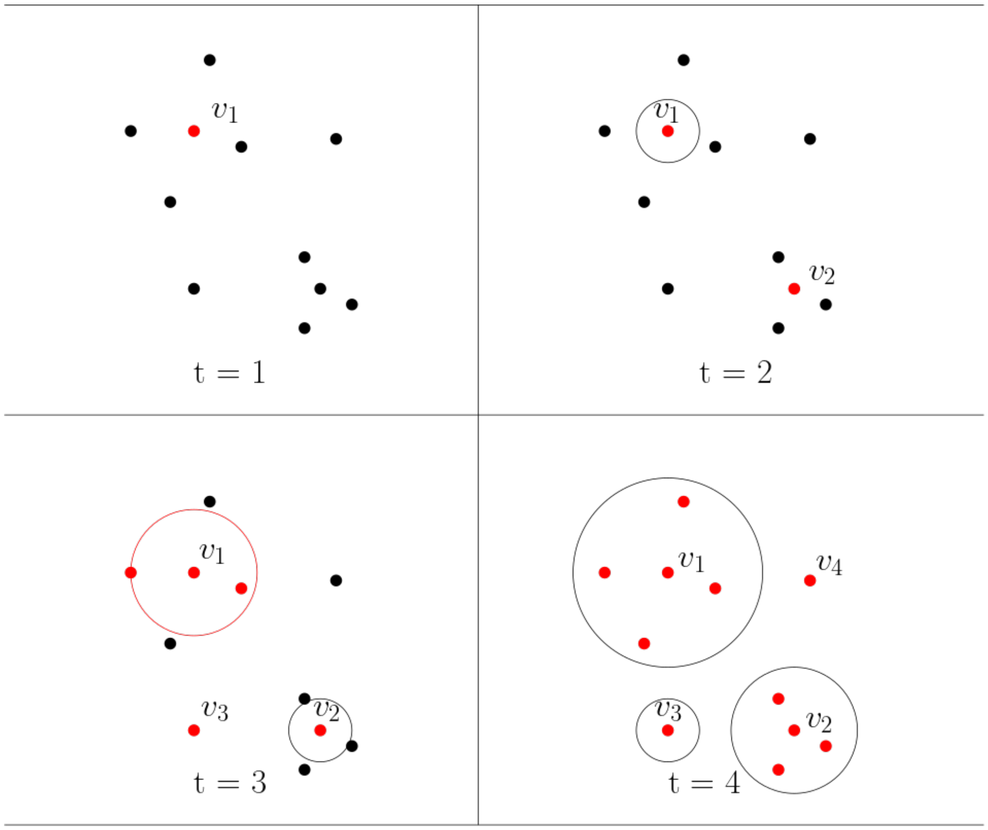

Fig. 2.0.1: Point Burning Process

Steps to burn points in the plane:

1. We pick v_1 as the initial burning source.

2. The fire from v_1 spreads to a radius of 1 unit, burning all points within that distance. At the same time, we pick v_2 as the second burning source.

3. The fire from v_1 reaches a radius of 2 units, burning 2 additional points within that region. Meanwhile, the fire from v_2 spreads to a radius of 1 unit. At this step, we pick v_3 as the third burning source.

4. The fire continues to spread from v_1, v_2, and v_3, burning all points except one. Finally, we pick that last point as v_4 to complete the burning process.

The point burning problem can be mathematically modeled as follows [19]:

Euclidean Burning Problem

Input: A set S of points in the $\mathbb{R}^2$ plane

Output: The minimum number of steps required to burn all the points of S.

This leads to the decision problem formulation [19]:

Decision Problem of Euclidean Burning

Input: A set S of points in the $\mathbb{R}^2$ plane and an integer b.

Question: Can all the points of S be burned in b steps?

This decision problem has been proven to be NP-Complete [19]. The complexity arises from the challenge of choosing optimal points to burn at each step. The point burning problem can be reduced from the path forest burning problem, which has already been proven to be NP-hard [4].

The point burning process can be visualized as covering the set of points with disks of increasing radii, starting from radius 0, then 1, 2, and so on. This interpretation allows us to draw a connection between the point burning problem and the minimum dominating set problem [19].

Many of the approximation algorithms use the concept of dominating set [16, 19, 26]. In this context, the minimum dominating set can be thought of as the minimum set of disks required to cover all the points in the plane [24]. But in the minimum dominating set, the radii of all disks are constant. By covering those constant radii disks with the disks of varying radii, we can use them for burning. The centers of these disks represent the burning sources, and the goal is to ensure that all the points fall within the regions covered by these disks. The point burning problem then becomes a task of finding the minimum number of such disks (or burning sources) necessary to cover all disks of the minimum dominating set.

2.1 Relationships to Dominating Set

A dominating set for a graph is defined as a set of vertices such that every vertex is either part of the dominating set or has at least one neighbor in the dominating set [24]. When we aim to minimize the number of vertices in this set, we obtain the minimum dominating set problem. This set ensures all vertices are covered with the minimum number of elements in the set and no smaller set can fulfill this condition [24].

To apply the concept of the dominating set to the points in the Euclidean plane, we first need to transform the set of points into a graph. This transformation is done using the concept of a unit disk graph. In this approach, each point is imagined to have a unit disk centered at it. If these disks have diameter r, we denote the resulting graph as G_r. When two disks overlap or intersect, an edge is drawn between the corresponding points [19].

By converting the points into a graph using this method, we can then compute the minimum dominating set of the resulting graph. The points in this set correspond to the minimum number of disks necessary to cover all other points, analogous to burning all points in the Euclidean graph burning problem [19].

The problem of finding the minimum dominating set is known to be NP-hard [8], even for unit disk graphs. In 2006, Nieberg and Hurink [24] proposed a Polynomial-Time Approximation Scheme (PTAS) to find a $(1 + \epsilon)$ approximation algorithm for the minimum dominating set in unit disk graphs. Their algorithm takes an undirected graph or a unit disk graph as input and returns a solution that approximates the minimum dominating set within a factor of $(1 + \epsilon)$ [24]. This result provides a near-optimal solution for the minimum dominating set problem.

Using the concept of unit disk graphs and the dominating set, various approximation algorithms have been developed to find the approximated burning number of points in the Euclidean plane [16, 19, 26].

2.2 Approximation Algorithms

2.2.1 (2+ϵ) Approximation Algorithm of Keil et al. [19]

In 2022, Keil et al. proposed a $(2 + \epsilon)$ approximation algorithm for burning a set of points in the plane. The algorithm works by incrementally guessing the burning number δ, for each guess, constructing a unit disk graph by placing disks of diameter δ centered at each point and finding the $(1+\epsilon)$-Approximatied minimum dominating set for that graph. The points in the dominating set are then chosen as the burning sources, and the fire spreads for extra δ steps, ensuring that all points are burned within the estimated burning number.

Algorithm A $(2 + \epsilon)$ Approximation Algorithm for Points in the Plane [19]

Input: $S \leftarrow$ set of points in $\mathbb{R}^2$, an integer $\delta \in \mathbf{Z}$
Output: $(2 + \epsilon)$-Approximation burning schedule
  $G_{\delta-1} \leftarrow$ unit disk graph with vertex set as S and edges where disks of radius $\frac{\delta}{2}$ intersects.
  $D_{\delta-1} \leftarrow (1 + \epsilon)$ approximation of minimum dominating set of $G_{\delta-1}$
  if $\frac{|D_{\delta-1}|}{(1+\epsilon)} > \delta$ **then**
    Return Bad-Guess
  else
    Return δ
  end if

Theorem 3. *(Keil et al., 2022, [19]) Algorithm A gives an $2 + \epsilon$ approximation.*

Proof. Here, we sketch the proof of [19]. For a set S of n points in the Euclidean plane, we first guess the burning number δ such that $1 \leq \delta \leq n$. For each δ, we construct the unit disk graph $G_{\delta-1}$, where the diameter of each disk is $\delta - 1$. Using the PTAS for the minimum dominating set, we compute an approximate dominating set $D'_{\delta-1}$ of $G_{\delta-1}$, with an approximation factor $(1+\epsilon)$ [24]. The set $D'_{\delta-1}$ contains the centers of disks that cover all points in the set S. Let the actual burning number of the set be δ^* the optimal number of steps needed to burn all points in S. Since $D'_{\delta-1}$ is a $(1+\epsilon)$ dominating set, we have:

$$\delta^* \geq |D_{\delta-1}|$$

where $D_{\delta-1}$ is the minimum dominating set for $G_{\delta-1}$.

If δ^* was smaller than $|D_{\delta-1}|$, it would imply that all points in S could be covered by disks of radius $\delta^* - 1$, contradicting the assumption that $D_{\delta-1}$ is the minimum dominating set. Hence, $\delta^* \geq |D_{\delta-1}|$ holds.

For each guessed value δ, the algorithm checks the condition: $\frac{|D'_{\delta-1}|}{(1+\epsilon)} \leq \delta$

If this condition is satisfied, the algorithm stops, and we can conclude that the optimal burning number $\delta^* \geq \delta$. Once the algorithm identifies the smallest δ satisfying the condition, the vertices in $D'_{\delta-1}$ are chosen as the burning sources. The fire spreads for $\delta - 1$ steps from each source, ensuring that all points within the range are burned. Since every point in S is either in the dominating set $D'_{\delta-1}$ or within distance $\delta - 1$ from a vertex in $D'_{\delta-1}$ all points in the set will be burned. The total number of steps taken by the algorithm to burn all points is:

$$|D'_{\delta-1}| + (\delta - 1)$$

Since $|D'_{\delta-1}| \leq (1+\epsilon)|D_{\delta-1}|$ and $\delta^* \geq |D_{\delta-1}|$, the total number of steps is bounded by:

$$(1+\epsilon)\delta^* + \delta^* = (2+\epsilon)\delta^*$$

Thus, the algorithm provides a $(2+\epsilon)$ approximation for the burning number. □

2.2.2 (1.96296+ϵ)-Approximation Algorithm of Gokhale et al. [16]

In 2022, Gokhale et al. proposed an improved algorithm for point burning. Instead of burning each source for a fixed number of steps as in previous approaches, they proposed to choose extra burning sources in those steps to burn the remaining points in a fewer number of steps.

Similar to the previous algorithm, this improved approach also begins by guessing the burning number, δ. For each guessed δ, the unit disk graph is generated the same way, and the minimum dominating set is calculated. The vertices in this dominating set are burned, but instead of burning for the additional δ steps, they burn for $\frac{26}{27}\delta$ steps. During the first $\frac{13}{27}$ steps, additional burning sources are selected, and as these sources burn for $\frac{13}{27}\delta$ steps, they cover the remaining region. This strategy helps optimize the process, ensuring efficient coverage of all points while minimizing the total burning steps.

The algorithm is based on a simple lemma:

Lemma 4. *(Gokhale et al., 2022, [16]) If we have two circles, C_1 with radius $r_1 = 1$ and C_2 with radius $r_2 = \frac{26}{27}$, consider the annulus defined by them. If we divide the annulus into 13 equal parts, then one disk of radius $\frac{13}{27}$ can cover one entire area.*

Figure 2.2.1 illustrates an annulus formed by two concentric circles with radii 1 and $\frac{26}{27}$, respectively. This annular region can be divided into 13 equal parts. If we select a point within any of these regions, then a disk of radius $\frac{13}{27}$ centered at that

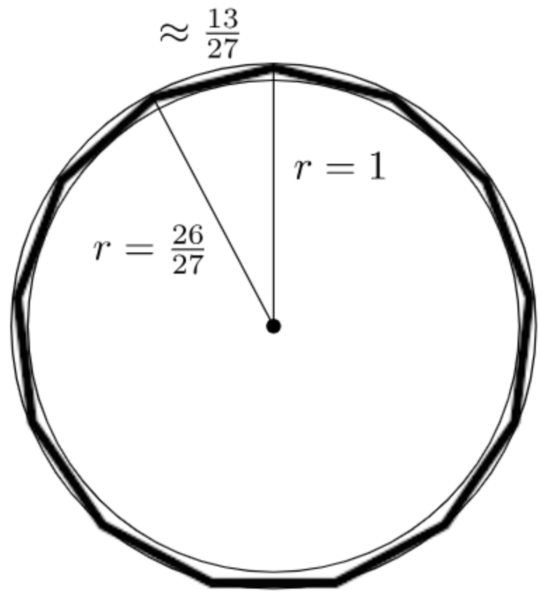

Fig. 2.2.1: Regular Polygon inscribed in annulus of circles with radius 1 and radius $\frac{26}{27}$

point will cover the entire corresponding section of the annulus. This concept ensures that by strategically choosing the additional burning sources in these sections, we can effectively reduce the number of steps to burn that entire region.

Theorem 5. *(Gokhale et al., 2022, [16]) We can modify algorithm A to get a (1.96296 + ϵ) approximation algorithm.*

Proof. We sketch the proof of [16]. The algorithm begins by guessing the burning number δ, just as in the previous approach. For each guessed δ a unit disk graph is constructed, and a minimum dominating set is identified. Instead of burning all the vertices for δ steps, the vertices in the dominating set are burned for $\frac{26}{27}\delta$ steps. During the first $\frac{13}{27}\delta$ steps, additional burning sources are chosen, ensuring that the remaining points are covered as the process continues. Consider the annulus formed between two circles of radii 1 and $\frac{26}{27}$ which can be divided into 13 equal parts. By selecting a point in each of these regions, a disk of radius $\frac{13}{27}$ is sufficient to cover the entire corresponding section of the annulus. This ensures that the additional burning sources cover the remaining unburned points efficiently. After the minimum number δ is returned by the algorithm, the burning process is carried out using the strategy described, which takes a total of $\frac{53}{27}\delta$ steps. Since the optimal burning number is at least δ, the ratio of the algorithm's steps to the optimal solution is $\frac{53}{27} \approx 1.96296$, giving a 1.96296-approximation. This proves that the algorithm efficiently burns all the points in the plane within $(1.96296+\epsilon)$ δ steps, providing the desired approximation. □

2.2.3 (1.944+ϵ)-Approximation Algorithm of Kamali and Shabanijou [26]

Kamali and Shabanijou's improvement on the point burning algorithm introduced a more efficient use of smaller disks to achieve a $(1.944+\epsilon)$-Approximation for burning all the points. To burn the points within $(1.944+\epsilon)\delta$ steps, the idea is to use the disks with radii 0, 1, up to $(1.944+\epsilon)\delta$ more efficiently.

Instead of relying on larger disks alone, the algorithm optimizes the use of smaller disks to cover areas that are not reachable by the larger disks in earlier steps. As in the previous algorithm, the majority of the points are covered with larger disks, but now there is a more refined strategy for the remaining areas. Unlike the previous algorithm where the annulus was divided into 13 equal parts, this new approach allows for dividing the annular region into varying numbers of sections depending on the situation. This flexibility helps ensure that the additional burning sources use smaller disks more efficiently to cover the remaining points.

Theorem 6. *(Kamali and Shabanijou, 2023, [26]) Algorithm B is $(1.944+\epsilon)$ approximation.*

Proof. Here, we sketch the proof of [26]. We begin by selecting a small value for ϵ and computing $\epsilon' = \frac{\epsilon}{1.944}$. This allows us to apply an approximation algorithm for the unit disk cover problem that returns a $(1+\epsilon')$ approximate solution. The solution, denoted by U', gives an approximation for the unit disk cover. The cardinality of U'

Algorithm B $(1.944+\epsilon)$ Approximation Algorithm

Input: $\delta \leftarrow$ minimum integer for which A does not return Bad-Guess, arbitrary small number $\epsilon \in \mathbb{R}$
Output: $(1.944+\epsilon)$-Approximation Schedule
$\epsilon' \leftarrow \frac{\epsilon}{1.944}$
$h^* \leftarrow$ cardinality of $(1+\epsilon')$ approximation of minimum dominating set $D_{\delta-1}$
$g^* \leftarrow$ smallest integer such that $g* \geq h*$ and $g^* < h^* + 10^4$
$D \leftarrow$ set of disks of radii $\{0, 1, \ldots, 1.944g^*\}$
Use disks of radii $[g^*, 1.944g^*]$ to cover $0.944g^*$ disks out of h^*
Use disks of radii $[0.944g^*, g^*)$ to cover remaining disks out of h^*
for the remaining $0.056g^*$ annuli,
for $7 \leq k \leq 16$ **do**
  divide annulus into k regions
  Use disks of radii $[2\sin(\frac{2\pi}{k}), 2\sin(\frac{2\pi}{k-1}))$ to cover $\lfloor \frac{\text{no. of disks}}{k} \rfloor$
end for

is denoted by h^*, representing the number of disks required to cover the points in the set. Assume h^* is arbitrarily large. We define g^* as the smallest integer no smaller than h^*, where $g^* < h^* + 10^4$.

We now attempt to cover all points using disks with radii from 0 to $1.944g^*$. Out of these $1.944g^*$ disks will have radii at least g^*. The remaining disks are classified into several groups, with class 6 consisting of disks with radii at least $0.944g^*$ (totalling $0.056g^*$ disks). These disks are centered at the remaining unit disk cover centers. For the remaining annular regions (defined between circles of radii 1 and 0.944), we use smaller disks from classes 7 to 16. Each class of disks is determined by dividing the annular region into k parts where $7 \leq k \leq 16$. For each class k disks with radii between $2\sin(\frac{2\pi}{k})$ and $2\sin(\frac{2\pi}{(k-1)})$ are used to cover the annuli. By doing so, each part of the annular region is covered with disks of an appropriate radius, as illustrated in Figure 2.2.2. The disks of class 6 cover the central region, while disks from classes 7 to 16 cover the remaining annuli. The total number of disks used across all classes exceeds $0.056g^*$, ensuring that all points in U' are fully covered. To burn all the points in the graph, we require at least $\frac{h^*-1}{1+\epsilon'}$ disks. Since $h^* > g^* - 10^4$, we conclude that at least $\frac{g^*-10^4}{1+\epsilon'}$ steps are required to burn all points. The total burning process uses disks with radii 0 to $1.944g^*$, so the total number of steps required is

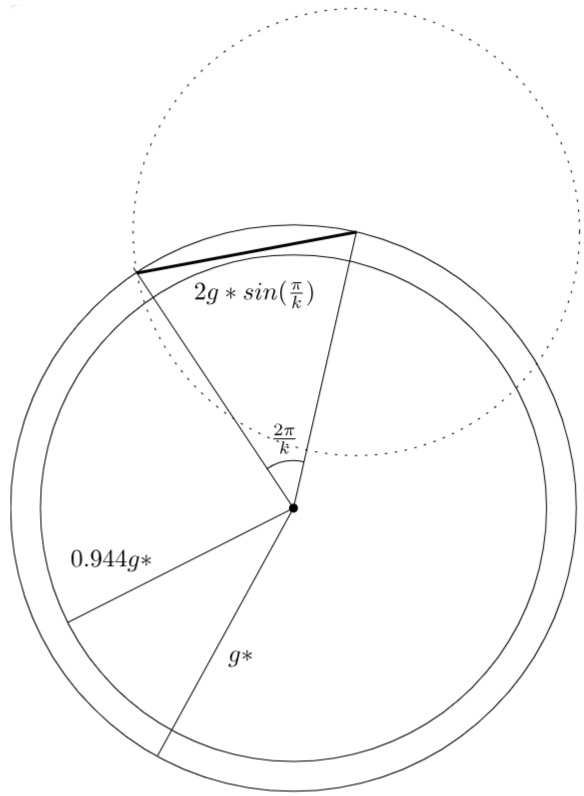

Fig. 2.2.2: Dividing the annulus of circles with radius 1 and 0.994 into k parts

$1.944g^*$. The approximation ratio for the algorithm is $\frac{\lfloor 1.944g^* \rfloor}{(g^*-10^4)(1+\epsilon')}$, which converges to $1.944(1+\epsilon') = 1.944+\epsilon$ for large g^*. Thus, the algorithm provides a $(1.944+\epsilon)$-Approximation for the point burning problem. □

Table 2.2.1 provides a breakdown of the number of disks in U' that are covered by each class of disks.

Class	Radius Range	No. of disks in the class	Partially covered disks of U'
1	$[g^*, 1.944g^*]$	$0.944g^*$	$0.0944g^*$
6	$[0.944g^*, g^*)$	$0.056g^*$	$0.056g^*$
7	$[0.8678g^*, 0.944g^*)$	$0.0762g^*$	$> 0.01g^*$
8	$[0.7654g^*, 0.8678g^*)$	$0.1024g^*$	$0.0128g^*$
9	$[0.6841g^*, 0.7654g^*)$	$0.0813g^*$	$> 0.009g^*$
10	$[0.6181g^*, 0.6841g^*)$	$0.066g^*$	$0.0066g^*$
11	$[0.5635g^*, 0.6181g^*)$	$0.0546g^*$	$> 0.0049g^*$
12	$[0.5177g^*, 0.5635g^*)$	$0.0458g^*$	$> 0.0038g^*$
13	$[0.4787g^*, 0.5177g^*)$	$0.039g^*$	$0.003g^*$
14	$[0.4451g^*, 0.4787g^*)$	$0.0336g^*$	$0.0024g^*$
15	$[0.4159g^*, 0.4451g^*)$	$0.0292g^*$	$> 0.0019g^*$
16	$[0.3902g^*, 0.4159g^*)$	$0.0257g^*$	$> 0.0016g^*$

Table 2.2.1: The number of disks with radius g^* covered by different groups of disks [26]

CHAPTER 3

Graph Burning - Classic Version

The increased usage of social media has resulted in the rapid growth of communication which motivated us to study the spread of information in the networks. This phenomenon, where news or rumors spread rapidly, can be studied mathematically with the concept of graph burning, suggested by Anthony Bonato in 2014 [4]. In this model, the vertices of the graph represent users in the network and the edges represent the connections between them. The process shows how the information spreads in real-world networks, such as social media, where a piece of information travels from user to user.

Graph burning is a discrete-time process on a graph $G(V, E)$, where initially all vertices are unburned. At each time step, a new vertex is selected out of the unburned vertices and it is burned, with previously burned vertices spreading the fire to their neighbors. The goal is to burn all the vertices of the graph in the fewest steps, simulating the spread of information. The steps required to burn the graph G is called its burning number and is denoted by $b(G)$. The main challenge of the problem is to find the optimal sequence of vertices to burn. The problem is proven to be NP-Complete [2].

Over time, various approximation algorithms have been proposed to find the approximated burning number [5]. Moreover, various upper bounds have been calculated to find the maximum number of steps required to burn any graph [1, 3, 4]. Various research has studied the graph burning number for different types of graphs such as binary trees [10], disjoint paths [5], spider graphs [9] and so on.

The classic version of graph burning problem has been studied for both undirected and directed graphs [18]. For directed graphs, the fire only spreads to the outgoing neighbors. That results in different bounds than its undirected counterpart [18].

Along with the introduction of the graph burning problem, Anthony Bonato [4] also proposed a famous conjecture that says that for any graph G of order n, $b(G)$ is at most $\sqrt{n}$ as discussed in 1. The conjecture still remains unproved.

Nonetheless, the conjecture has been confirmed asymptotically, showing that a graph of n vertices can be burned in $(1 + o(1))\sqrt{n}$ steps [25]. Further research continues advancements in the field resulting in tighter bounds and progresses to prove the conjecture.

3.1 Properties of Burning Number by Bonato et al. [4]

Bonato et al. discussed some important properties of burning a graph. We summarise them in this section.

Proposition 1. *For a graph $G = (V, E)$ with a burning sequence $(x_1, x_2, \ldots, x_k)$, the vertices of the graph satisfy: $N_{k-1}[x_1] \cup N_{k-2}[x_2] \cup \cdots \cup N_0[x_k] = V(G)$, where $N_r[v]$ denotes the rth neighborhood of vertex v, which is the set of vertices at a distance at most r from v.*

Since we pick a new burning source after the fire spreads in each step, the following property holds:

Proposition 2. *For a graph $G = (V, E)$ with a burning sequence $(x_1, x_2, \ldots, x_k)$ and for all $1 \leq i < j \leq k$, it holds that: $d(x_i, x_j) \geq j - i$, where $d(x_i, x_j)$ denotes the distance between vertices x_i and x_j.*

Proof. Assume, for contradiction, that $d(x_i, x_j) < j - i$. Since $i < j$, vertex x_i was burned at step i, and x_j was selected to be burned at step j. The fire from x_i takes exactly $j - i$ steps to reach x_j. However, if $d(x_i, x_j) < j - i$, then x_j would already have been burned by step j, meaning it could not have been selected as a burning source at that step. This contradicts the assumption that x_j was burned at step j, thus proving the claim. □

A spanning subgraph of a graph contains all the vertices but it need not contain all the edges of the original graph. The burning number of a graph is upper bounded by the burning number of its spanning subgraph. This holds because a spanning subgraph contains all the vertices of the original graph but potentially fewer edges, making it easier or equally challenging to spread the fire in the original graph.

Theorem 7. *Let H be a spanning subgraph of G. Then, the burning number of G satisfies: $b(G) \leq b(H)$.*

Proof. If a burning sequence can burn all the vertices of H, then it can also burn all the vertices of G, since H is a subgraph of G. Therefore, the burning number of G is at most the burning number of H. □

The burning number of a graph is closely related to the burning number of its spanning tree. More precisely, the burning number of a graph is same as the minimum burning number among its spanning trees.

Theorem 8. *For a graph G, $b(G) = min\{b(T) : T \text{ is a spanning tree of } G\}$*

Proof. Since a spanning tree is also a spanning subgraph of G, we know that if T_m is the spanning tree with the minimum burning number, then $b(G) \leq b(T_m)$.

Now, consider a rooted tree partition of G, consisting of trees $T_1, T_2, \ldots, T_k$, where T_k has height 0, T_{k-1} has height 1, and so on until T_1 has height $k-1$. In this case, $b(G) = k$. A rooted tree partition is a partition of a rooted tree such that the union of the all trees gives back the entire vertex set of the tree. If we connect all the rooted trees without inducing a cycle, the resulting tree T is a spanning subtree. By spanning subtree, we mean that that it contains all the vertices of the original tree but not all the edges. By burning the trees in the order $T_1, T_2, \ldots, T_k$, the tree T can be burned in k steps, implying $b(T) \leq k = b(G)$.
Combining both results, we conclude that $b(G) = b(T)$. □

The burning number of a graph is also related to the burning number of its isometric subgraph. An isometric subgraph preserves the shortest path distances between vertices from the original graph. Since distances are preserved, the fire spreads in the same manner in both the original graph and its isometric subgraph.

Theorem 9. *For a graph G and its isometric subgraph H, $b(H) \leq b(G)$.*

Proof. Since $H = (V_H, E_H)$ is an isometric subgraph of $G = (V_G, E_G)$, we know that $V_H \subseteq V_G$ and the distance between any two vertices $v_i, v_j \in V_H$ is the same as the distance between v_i and v_j in V_G.
Thus, we can use the same burning sequence for H as for G, implying that $b(H) \leq b(G)$.

□

The process of burning a path is straightforward. We select burning sources such that they burn an odd number of vertices 1, 3, 5, and so on. In this way, the fire spreads efficiently.

Theorem 10. *For a path P_n, $b(P_n) = \lceil \sqrt{n} \rceil$.*

Proof. The burning sources chosen at times $t = 1, 2, 3, \ldots, k$ will burn $1, 3, 5, \ldots, 2k-1$ vertices of the path. We can burn the entire path if the sum of the burned vertices is greater than or equal to n (the total number of vertices).

$$\sum_{i=1}^{k}(2i-1) = k^2 \geq n,$$

hence,

$$k \geq \lceil \sqrt{n} \rceil.$$

Thus, $b(P_n) = \lceil \sqrt{n} \rceil$. □

The same process applies to burning a cycle as well.

Theorem 11. *For a cycle C_n, $b(C_n) = \lceil \sqrt{n} \rceil$*

Proof. The proof is the same as for Theorem 10. We can remove an edge from the cycle C_n, reducing it to a path P_n, and apply the same burning strategy. □

The burning number of a graph can also be related to the graph's radius and diameter. The radius of a graph is the minimum distance from a central vertex to all other vertices in the graph. The radius of the tree is defined in the same way. The diameter of a graph is the greatest distance between any two vertices in the graph.

Theorem 12. *For a graph G with radius r and diameter d, $\lceil \sqrt{d+1} \rceil \leq b(G) \leq r+1$*

Proof. The eccentricity of a vertex v in G is the greatest distance from v to any other vertex in the graph. If we burn the vertex with minimum eccentricity, the fire will spread to all vertices in r steps, so $b(G) \leq r+1$.
Now, consider a path P between two vertices u and v such that $d(u,v) = d$. The path P is an isometric subgraph with $d+1$ vertices. To burn a path with $d+1$ vertices, we need $\lceil \sqrt{d+1} \rceil$ steps. Since $b(P_{d+1}) \leq b(G)$, we have $\lceil \sqrt{d+1} \rceil \leq b(G)$. □

The burning number is also related to the k-distance dominating number of a graph. A dominating set in a graph is a set of vertices such that every vertex in the graph is either in the dominating set or adjacent to a vertex in the dominating set. The k-distance dominating set generalizes this idea by allowing vertices to be at most k distance away from any vertex in the dominating set.

Theorem 13. *For a graph G with $|V(G)| \geq 2$ and optimal burning number k, let $\gamma_{k-1}(G)$ be the $(k-1)$-distance minimum dominating set of G. Then, $\gamma_{k-1}(G) \leq k$.*

Proof. The value $\gamma_{k-1}(G)$ represents the cardinality of the $(k-1)$-distance dominating set, meaning that all vertices of G are within distance $k-1$ from the dominating set. Suppose $k < \gamma_{k-1}(G)$. All vertices of G are at most distance $k-1$ from some burning source, meaning that the set of burning sources forms a dominating set for G with respect to radius $k-1$. However, since $k < \gamma_{k-1}(G)$, this contradicts the fact that $\gamma_{k-1}(G)$ is the minimum size of a dominating set where every vertex is within distance $k-1$ of some dominating vertex. Therefore, the assumption $k < \gamma_{k-1}(G)$ must be false. □

Burning Number Decision Problem:

Input: a graph G and an integer $k \in \mathbb{R}$

Question: Is the burning number of the graph $b(G)$ is less than or equal to k?

Considering the burning number properties, we can conclude that the graph of n vertices which is the hardest to burn may be a path. Since adding more number of edges can increase the number of vertices burned per step. In the same way, the easiest graph to burn is a complete graph or a spider graph where the center vertex has $n-1$ neighbors. In both cases, the graph can be burned in 2 steps.

As mentioned before, the decision version is defined as above. Bonato et al. [2] proved that this decision problem is NP-Complete. They proved that the graph burning problem is polynomial time reducible from distinct 3-partition problem. This result holds even when G is planar or disconnected [2].

3.2 Approximation Algorithms

Since the graph burning problem is NP-complete, many approximation algorithms have been introduced over the past decade [5].

3.2.1 3-Approximation Algorithm of Bonato and Kamali [5]

Bonato and Kamali proposed a 3-approximation algorithm for burning a graph. The algorithm begins by guessing the burning number of the graph G, denoted by g. It then selects an arbitrary vertex as the first burning source and marks all vertices within a distance of $2g-1$ from it. This process is repeated, selecting new burning sources and marking their respective neighborhoods until all vertices in the graph are either burning sources or marked.

If the number of burning sources reaches g and the entire graph has not yet been burned, the guess for g is deemed incorrect, referred to as a Bad-Guess. This implies that the graph cannot be burned in g steps, and a larger guess for the burning number is needed. The pseudocode is given in algorithm C.

Lemma 14. *(Bonato and Kamali, 2019, [5]) If the number of centers becomes equal to g, then at least g steps are required to burn the graph G.*

Proof. When the number of centers becomes equal to g, there are g vertices with pairwise distance $2g-1$. To burn G in g steps, the maximum radius of any burning source can be at most $g-1$. Given that the pairwise distance between centers is $2g-1$, at least one source is needed for each center. Thus, at least g steps are required to burn the entire graph. □

Theorem 15. *(Bonato and Kamali, 2019, [5]) The algorithm C is a 3-approximation algorithm.*

Algorithm C 3-Approximation Algorithm for General Graph

Input: $G = (V, E) \leftarrow$ general undirected graph, g $\in \mathbf{N}$
Output: 3-Approximation Burning Schedule

```
unmarked ← V
centres ← []
while |unmarked| > 0 do
    v ← arbitrary vertex from unmarked
    c ← the closest centre from v
    if d(v, c) > 2g − 2 then
        centres = centres + v
        if |centers| ≥ g then
            return Bad-guess
        end if
    end if
    unmarked = unmarked \ v
end while
```

Proof. The algorithm iterates the value of g from $1, 2, \ldots$ until it finds the smallest g_m such that a valid burning sequence is achieved. For g_m, we burn the $g_m - 1$ centers in arbitrary order and allow the burning process to continue for another $2g_m - 2$ steps to ensure that the entire graph G is burned. This process takes $3g_m - 3$ steps. Since g_m is the smallest integer for which a valid burning sequence exists, the algorithm returns a Bad-Guess for $g_m - 1$, meaning the optimal burning sequence requires at least $g_m - 1$ steps.

Thus, the approximation ratio is given by:

$$\frac{3g_m - 3}{g_m - 1} = 3$$

$\square$

3.2.2 2.313-Approximation Algorithm of Martinsson [21]

Anders Martinsson proposed a greedy random algorithm to burn the graph and gave a 2.313-approximation ratio. In this, we choose a disk of random radius from the uniform distribution and place it on a random unburned vertex.

Algorithm D 2.313 Approximation Algorithm

Input: $G = (V, E), m \in \mathbf{Z}$
Output: 2.313-Approximation Burning Schedule
  $t \leftarrow 0$
  $V' \leftarrow V$
  while $v \in V'$ **do**
    $r \sim U[0, m]$
    Place disk of radius r centred on v
    $V' = V/N_r(v)$
    $t \leftarrow t + 1$
  end while

Theorem 16. *(Martinsson, 2023, [21]) The given algorithm D is a 2.313-approximation if* $m \geq \left(\frac{2}{1+e^{-2}}\right) b(G)$.

Proof. The algorithm is randomized. While there are any unburned vertices, it selects a burning radius from a uniform distribution over $[0, m]$. Let $B_0, B_1, \ldots, B_{b(G)-1}$ be the optimal burning schedule for G. The probability that the chosen vertex v is B_i for $i \in \{0, 1, \ldots, b(G) - 1\}$ is $1 - \frac{2i}{m}$. The expected value of the algorithm placing the burn center at B_i is $\frac{1}{1-\frac{2i}{m}}$. Thus, the algorithm places at most:

$$\sum_{i=0}^{b(G)-1} \frac{1}{1 - \frac{2i}{m}} \leq \int_0^{b(G)} \frac{1}{1 - \frac{2x}{m}}\, dx = \frac{m}{2} \log \frac{1}{1 - \frac{2b(G)}{m}}.$$

To burn G, we choose a value for m between $\frac{2}{1-e^{-2}} b(G)$ and $\frac{2}{1-e^{-2}} b(G) + O(\sqrt{m \log m})$. We then select a set of radii to cover G by running the algorithm. Let r be the smallest integer such that the set $(0, 1, \ldots, r-1)$ dominates the burning sequence. The algorithm uses m disks with probability at most $\frac{1}{m+1}$, and the corresponding burning sequence is dominated by $(0, 1, \ldots, m + O(\sqrt{m \log m} - 1))$. Therefore, $b(G) \leq m + O(\sqrt{m \log m})$ with probability at least $\frac{1}{2(m+1)}$. Repeating this procedure $\omega(m \log m)$ times ensures that $b(G) \leq (1 + \epsilon)m$ with high probability. $\square$

3.2.3 2-Approximation Algorithm for Trees of Bonato and Kamali [5]

The best-known approximation algorithm for trees till now was introduced by Bonato and Kamali. In this algorithm, after guessing the burning number g, we look for the vertex v with the highest level, that means the vertex with the maximum distance from the root, and then add the g^{th} ancestor of v in the list of centers. We mark the g^{th} neighbourhood of center as burned and repeat the process for the remaining unburned vertices.

Algorithm E 2-Approximation Algorithm for General Tree

Input: $T(V, E) \leftarrow$ general undirected tree, g $\in \mathbf{N}$
Output: 2-Approximation Burning Schedule
  unmarked $\leftarrow V$
  centres $\leftarrow$ [ ]
  while |unmarked| > 0 **do**
    $v \leftarrow$ vertex of highest level.
    if level of $v <$ g **then**
      centres $\leftarrow$ centres + s
    else
      centres $\leftarrow$ centres + g^{th}-ancestor of v
    end if
    unmarked $\leftarrow$ unmarked $\setminus N_g(v)$
    if |centres| $> g$ **then**
      return Bad-Guess
    end if
  end while

To calculate the approximation ratio, we use the concept of a g-site partition. A g-site partition is a set of at most g vertices such that all other vertices are within distance g from them. If the tree does not have a g-site partition, it implies that no set of g vertices with radius g can cover the tree. Therefore, $b(T) > g$.

Theorem 17. *(Bonato and Kamali, 2019, [5]) The algorithm E is a 2-approximation algorithm.*

Proof. Let T be an undirected tree, and let g be the minimum integer for which the algorithm returns the burning sequence. This means the algorithm produces a Bad-Guess for $g-1$, indicating that the tree cannot be burned in $g-1$ steps, as it does not have a $(g-1)$-site partition. Hence, $b(T) > g-1$.

The algorithm burns the g vertices it selects, and burns for an additional g steps, resulting in a total of $2g$ steps.

Therefore, the approximation ratio is:

$$\frac{2g}{g} = 2.$$

□

3.2.4 Farthest-first Algorithm of Garcia-Diaz et al. [14]

Garcia-Diaz et al. proposed a greedy method for burning a graph. The process begins by selecting an arbitrary vertex and adding it to the list of burning sources. In subsequent steps, the algorithm selects the vertex that is farthest from all the vertices already in the list of burning sources. They proved that selecting $\left(3 - \frac{2}{b(G)}\right)$ vertices using this method is sufficient to burn all the vertices in the graph, where $b(G)$ is the optimal burning number.

Lemma 18. *(Garcia-Diaz et al., 2022, [14]) Choosing $3 - \frac{2}{b(G)}$ vertices with this method is sufficient to burn all the vertices where $b(G)$ is the optimal burning number.*

Theorem 19. *(Garcia-Diaz et al., 2022, [14]) The algorithm F is a $3 - \frac{2}{b(G)}$ approximation, where $b(G)$ is the optimal burning number of the given graph G.*

Proof. In the algorithm F, we select the vertices that are at the maximum distance from the previously chosen burning sources. By Lemma 18, we know that the algorithm selects at most $3 - \frac{2}{b(G)}$ burning sources to ensure all vertices are burned. Hence, the approximation ratio of the algorithm is $3 - \frac{2}{b(G)}$. □

Algorithm F Farthest-First Algorithm

Input: $G = (V, E) \leftarrow$ an undirected graph
Output: $(3 - \frac{2}{b(G)})$-Approximation Burning Schedule
  Compute all pair-wise shortest distance
  $v \in V$ is an arbitrary vertex
  $S \leftarrow \{v\}$
  $B_{\text{previos}} \leftarrow \phi$
  $B_{\text{current}} \leftarrow \{v\}$
  while $B_{\text{current}} \neq V$ **do**
    $u \leftarrow argmax_{k \in V} d(k, S)$
    $S = S \cup \{u\}$
    $B_{\text{previous}} = B_{\text{current}}$
    $B_{\text{current}} = B_{current} \cup N(B_{\text{current}}\ B_{\text{previous}}) \cup \{u\}$
  end while

3.2.5 Directed Graph

The burning process for directed graphs was introduced by Remie Janssen [18]. In this variant, the burning process is similar to that of undirected graphs, but the key difference is that the fire only spreads to the out-neighbours of a vertex. This restriction changes the bounds of the burning process, because it limits the spread of the fire [18].

Finding the burning number of a directed graph is also NP-hard, and this can be proven in the same way as undirected graphs [18].

3.2.5.1 Multirooted Directed Tree of Gautam et al. [15]

Gautam et al. introduced a 3-approximation algorithm for multirooted directed trees.

The algorithm leverages the concept of the b-cutting process. The b-cutting process is defined as follows: remove vertices with in-degree 0 and out-degree 1, and repeat this process b times. This iterative approach helps in approximating the optimal burning number for directed trees.

The algorithm G shows the pseudocode. We guess the burning number b. In this

Algorithm G 3-Approximation Algorithm for Multirooted Directed Tree

Input: $T \leftarrow$ multirooted directed tree, an integer $b \in \mathbb{N}$
Output: 3-Approximation Burning Schedule

```
BS ← φ
BS′ ← φ
while |V(T)| > 0 do
    T′ ← (b − 1)-cutting of T
    for v ∈ V(T′) do
        if D⁺(v) = 0 then
            if D⁻(v) ≤ 1 then
                BS ← BS ∪ {v}
            else
                BS′ ← BS′ ∪ {v}
            end if
            T ← T \{u : d(u,v) ≤ b and u ∈ V(T)}
        end if
    end for
    if |BS| > b or |BS′| > b then
        Return Bad-Guess
    else
        Return (BS, BS′)
    end if
end while
```

algorithm, after performing $(b-1)$-cutting, we divide the vertices of indegree 1. If the vertex has an outdegree at most 1 then we put it in the list BS, otherwise, we put it in the list BS'. We then mark all the vertices within distance b from any vertex of list BS and BS' as marked.

Lemma 20. *(Gautam et al., 2023, [15]) In algorithm G, if $|BS| > b$ then $b(T) > b$.*

Proof. BS is a list of vertices with in-degree 0 and out-degree 1 after performing the $b-1$-cutting. It means BS is a list of vertices which are either roots or have a tail of length $b-1$. One burning source can burn at most one vertex in the list BS. So, if $|BS| > b$ then we need at least b burning sources. □

Lemma 21. *(Gautam et al., 2023, [15]) In algorithm G, if $|BS'| > b$ then $b(T) > b$.*

Proof. BS' is a list of vertices with in-degree > 1 means all the vertices of BS' will have more than one parent node. If $|BS'| > b$ then there will be at least $b+1$ parent nodes since two vertices of BS' cannot share more than one parent as that would result in a cycle. To burn $b+1$ parents with no common ancestors, we need at least $b+1$ burning sources, so if $|BS'| > b$ then $b(T) > b$. □

To burn a tree, we begin by selecting burning sources from the sets BS and BS' in an arbitrary order. Then we allow the fire to spread for an additional b steps. Since all the vertices are within distance at most b from the burning sources, by allowing the fire to propagate over these b steps, the entire tree will be burned completely.

Theorem 22. *(Gautam et al., 2023, [15]) The algorithm G is a 3-approximation algorithm.*

Proof. Here, we present the sketch of proof of [15]. We iteratively apply the algorithm G to find the smallest integer b for which the algorithm G does not return a Bad-Guess. Since b is the smallest integer meaning the algorithm G returns a Bad-Guess for $b-1$ because either $|BS| > b-1$ or $|BS'| > b-1$. In either case, we can say that $b(T) > b-1 \implies b \leq b(T)$ where $b(T)$ is the optimal burning number.

By burning all the vertices in the lists BS and BS' and letting them burn for another b steps, we can burn all the vertices of the tree T. Since the algorithm G does not return a Bad-Guess, it means $|BS| \leq b$ and $|BS'| \leq b$, so the total number of steps required is $|BS|+|BS'|+b \leq 3b$. Therefore, the approximation ratio is 3. □

3.2.5.2 Single Rooted Directed Tree of Gautam et al. [15]

Gautam et al. suggested a modified version of the algorithm G that achieves a 1.905-approximation for single-rooted directed trees.

We can use the same algorithm G to burn single-rooted trees as well. In this case, $BS' = 0$, so we obtain $|BS| + b = 2b$, resulting in a 2-approximation algorithm. The main improvement is merging two burning sources that have the lowest common ancestor within a certain distance.

Lemma 23. *(Gautam et al., 2023, [15]) If the number of merge trees is less than* $0.095b$*, where* b *is an input integer, then we cannot burn the tree in* b *steps.*

We can use the lemma 23 to improve the approximation ratio from 2 to 1.905.

Theorem 24. *(Gautam et al., 2023, [15]) The algorithm H is a 1.905-Approximation Algorithm.*

Proof. Here, we sketch the proof of [15]. The algorithm H either returns a Bad-Guess or a list of burning sources. Let b be the smallest integer for which the algorithm returns a valid burning sequence. This implies that the algorithm will return a Bad-Guess for $b-1$. Initially, we have the list BS where $|BS| < b$. If we let these burning sources burn for b steps, we can cover the entire tree. However, merging at least $0.095b$ trees results in a merged tree with a height of at most $1.81b$. We use the

Algorithm H 1.905-Approximation Algorithm

Input: $T \leftarrow$ single-rooted directed tree, an integer $b \in \mathbb{N}$
Output: 1.905-Approximation Burning Schedule

```
BS ← BS received from the algorithm G
if |BS| > b then
    Return Bad-Guess
else
    merges ← 0
    new_centers ← ∅
    for u, v ∈ BS and u ≠ v do
        k ← lowest-common-ancestor of u and v
        if d(k, u) ≤ 0.81b and d(k, v) ≤ 0.81b then
            BS ← BS \ {u, v}

        end if
    end for
    if merges ≥ 0.095b then
        Return Bad-Guess
    else
    Return (BS,
    end if
end if
```

set of burning radii $\{0, 1, \ldots, 1.905b\}$ to cover the entire tree. The sources with radii $\{b, b+1, \ldots, 1.81b\}$ cover the unmerged trees, as the number of unmerged trees will be at most $b - 2 \times 0.095b = 0.81b$. The remaining radii set $\{1.81b + 1, \ldots, 1.905b\}$ is used to cover the merged trees, where the number of sources in this set is $1.905b - 1.81b = 0.095b$. Hence, the tree can be burned in $1.905b$ steps, while the optimal burning number is at least b since $b - 1$ returns a Bad-Guess. This gives an approximation ratio of 1.905. □

3.3 Combinatorial Bounds

It is challenging to find the minimum number of steps required to burn the graph [2]. That motivates us to find the upper bound for the graphs to observe the maximum number of steps required for burning.

3.3.1 General Graphs

Bessy et al. [3] proved that $b(G) \leq \sqrt{\frac{12n}{7}}$. Over the decade, a lot of efforts were made to improve this bound. Bestide et al. [1] provided a simple proof for an improved bound of $\sqrt{\frac{4}{3}n} + 1$.

Lemma 25. *(Bestide et al., 2021, [1]) Given a tree T and a nonempty finite set of integers $R \subseteq \mathbb{N}$, there exists a subtree T' and an integer $r \in R$ such that*

1. *$T \setminus T'$ is connected,*
2. *radius(T') $\leq r$,*
3. *$|V(T')| \geq r + \frac{|R|}{2}$.*

Proof. We present the sketch of proof of [1]. Let P be the longest path in T with vertices $(v_0, v_1, \ldots, v_p)$. Let v_p be the new root and let T_i denote the subtree rooted at vertex v_i. Define $\phi(i)$ as the maximum index such that for $j \in \{0, 1, \ldots, p\}$, T_j is part of the i-neighborhood of v_i. This means

$$\phi(i) = \max\{j : T_j \subseteq N_i(v_i)\}.$$

Clearly, $i \leq \phi(i) \leq 2i$. If for some $r \in R$, $\phi(r) = p$, then we have proved the lemma.

Assume instead that for every $r \in R$, $\phi(r) < p$. If $T_{\phi(r)}$ has more than $r + \frac{|R|}{2}$ vertices, then we also get the expected result. So assume that for every $T_{\phi(r)}$ where $r \in R$, the number of vertices $|V(T_{\phi(r)})| < r + \frac{|R|}{2}$. Consider R' to be the set of the maximum N elements, where $N = \frac{|R|}{2}$. For $r' \in R'$, we know that $r' \geq \frac{|R|}{2} - 1$ and $\phi(r') < 2r'$, because we assumed $\phi(r) < r + \frac{|R|}{2} - 1$. Since $\phi(r') < 2r'$, there will be a vertex $x_{r'}$ such that $x_{r'} \in T_{\phi(r')+1} \setminus T_{\phi(r')}$ and $d(x_{r'}, v_{r'}) = r' + 1$ as shown in the figure 3.3.1. Each $r' \in R'$ has a distinct $x_{r'}$. Also, if $\phi(r'_i) > \phi(r'_j)$, then $x_{r'_j} \in T_{\phi(r'_i)}$. Let the elements of R' be $r'_1 < r'_2 < \ldots < r'_N$ and m be the minimum index such that

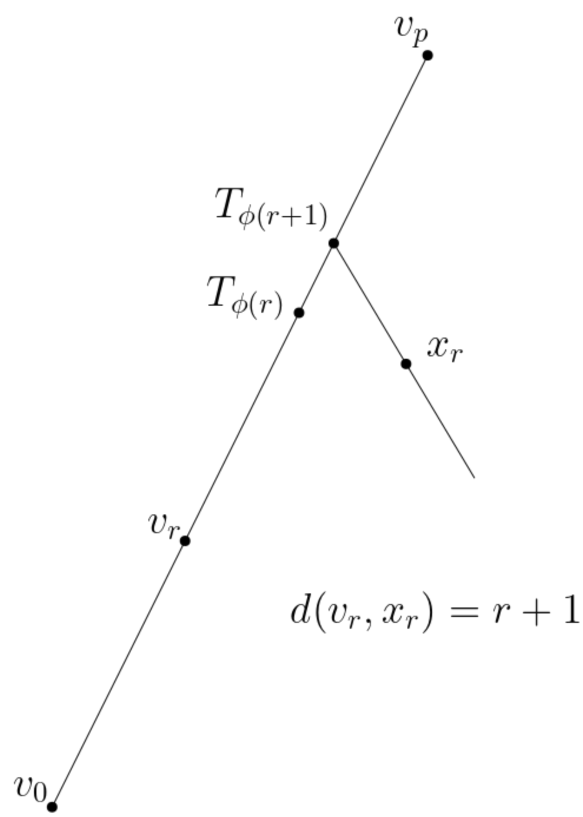

Fig. 3.3.1: There exist $x_{r'}$ for v_r

$\phi(r'_m) \geq \phi(r'_N)$. We calculate the number of vertices in $T_{\phi(r'_m)}$. We have

$$|V(T_{r'_m})| \geq m + r'_N \geq m + r'_m + (N - m) \geq r'_m + N \geq r'_m + \frac{|R|}{2}.$$

□

R is an independent set of integers from tree T but lemma 25 shows that we can always find an element r from R such that if we burn a vertex in the subtree $T' \subset T$ and let it burn for r steps then we will be able to burn entire subtree T'.

To burn a graph G within p steps using burning sources with radii $\{1, 2, 3, \ldots, p\}$, and to improve the approximation ratio, we can use a lemma 25 repetitively to find a spanning subtree with an appropriate size.

Theorem 26. *(Bestide et al., 2021, [1]) For any connected graph G with n vertices,*

$b(G) \leq \sqrt{\frac{4}{3}n} + 1$.

Proof. We use Lemma 25 to prove this theorem. Let T be a spanning tree of graph G and let p be the integer we guess for the burning number. Let R be the set of the first p integers. By applying Lemma 25 iteratively, we can say that T is not burnable in p steps unless

$$n > \sum_{i=0}^{p} i + \sum_{i=0}^{p} \frac{i}{2} = \frac{3p(p+1)}{4}.$$

Solving this equation gives us

$$p \leq \lceil \sqrt{\frac{4}{3}n} \rceil + 1.$$

□

Norin and Turcotte [25] showed that the burning number conjecture holds asymptotically.

Theorem 27. *(Norin and Turcotte, 2024,[25]) For a connected graph G with n vertices, $b(G) \leq (1 + o(1))\sqrt{n}$.*

To prove this theorem, the authors use the concept of metric trees to get benefits from the continuous and probabilistic settings. They prove that the converted metric tree can be covered with the disks of varying radii such that the cover is frugal meaning the sum of their radii will be minimal [25].

3.3.1.1 Our Contribution

We claim that the same result of theorem 26 can be achieved by a better approach.

In lemma 25, we are only considering the maximum $\frac{N}{2}$ members of the same R where R is the set of first N natural numbers. But for the rest of the smaller elements of set R, we claim that we can find a subtree T' such that we can either burn $2r + 1$

vertices or there exists x_r corresponding to the element r.

Theorem 28. *Given a tree T with the longest path $\{v_0, v_1, \ldots, v_l\}$ Let $R = \{1, 2, \ldots, p\}$ be the set of first p integers. For $r \in \{1, 2, \ldots, \frac{p}{2}\}$,*

- *either there exists x_r which is not on the longest path of the tree T*
- *or we can burn $2r + 1$ vertices.*

Proof. Its proof is straightforward. In the set R, we check for an element r such that $\phi(r) = 2r$. In that case, if we burn the v_r, in r steps we will be able to burn at least $2r + 1$ vertices.

In case, no such r exist, means for every $r \in R$, $\phi(r) < 2r$. In this case, we will find separate x_r for every $r \in R$. So, we will be able to burn $r + |R|$ vertices.

□

We can use this result to find elements of R such that it either burns $2r + 1$ or $r + |R|$ vertices. Each element r of R corresponds to the burning source with radius r. If we calculate the minimum number of vertices possible to burn this way, we know that for $r \leq \frac{p}{2}$, $2r + 1$ will be less than $r + |R|$. If we sum them, we get

$$\sum_{1}^{\frac{p}{2}} 2r + 1 + \sum_{\frac{p}{2}}^{p} r + |R|$$

This equals to $\frac{3p^2}{4} + O(n)$ which gives us the same bound.

3.3.2 General Trees

Many studies focus on determining the bounds for the burning number of trees [10, 20, 23]. These results are directly applicable to graphs by considering their spanning trees.

Theorem 29. *(Das et al., 2023, [10]) Let T be a tree with n vertices, out of which n_2 vertices have degree 2. Then, $b(T) \leq \sqrt{n + n_2 + 8} - 1$.*

Sandip Das et al. proved this bound using mathematical induction. For further details, please refer to [10].

A Homeomorphically Irreducible Tree is a tree where there is no vertex of degree 2. Murakami proved burning conjecture for them [23].

Theorem 30. *(Murakami, 2023, [23]) For a Homeomorphically Irreducible Tree (HIT) T with n vertices, $b(T) \leq \lceil \sqrt{n} \rceil$.*

Proof. We sketch the proof of [23]. Let $T_u(uv)$ be the subtree including vertex u when the edge uv is removed. In an HIT, we can always find a vertex x with neighbors $v_1, v_2, \ldots, v_k = y$ such that $|T_x(xy)| \geq 2\sqrt{n}-1$ and for $i \in \{1, 2, \ldots, k-1\}$, $|T_{v_i}(xv_i)| < 2\sqrt{n} - 1$. So if we burn x in step 1, after $\lceil \sqrt{n} \rceil$ steps, all vertices in $T_x(xy)$ will be burned. The remaining vertices will be

$$n - |T_x(xy)| \leq n - 2\sqrt{n} + 1 \leq (\sqrt{n} - 1)^2.$$

We can use induction to burn the tree in $\sqrt{n}$ steps. □

It is obvious that we can convert any tree to HIT if we add a pendant vertex to the vertices of degree 2. The pendant vertex is a vertex with degree 1.

Theorem 31. *(Murakami, 2023, [23]) For a tree T with n vertices, out of which d vertices are of degree 2, we can burn T in $\sqrt{n + d}$ steps.*

Proof. We connect a pendant vertex to all d vertices of degree 2 to convert T into an HIT. Using the result from Theorem 20, we can say that we will need $\sqrt{n+d}$ steps to burn T. $\square$

3.3.3 Graphs with Minimum Degree k of Martinsson [20]

Anders Martinsson proposed results related to the burning number for graphs with a minimum degree k. These results focus on finding a suitable connected subgraph for burning.

Theorem 32. *(Martinsson, 2023, [20]) For a graph G with n vertices and minimum degree k, there exists a connected subgraph H with at most $3\lfloor \frac{n}{k+1} \rfloor - 2$ vertices such that all the vertices of G are within distance 2 from H.*

Proof. We sketch the proof of [20]. We start with an arbitrary vertex H_0 of G. Then, we iteratively add a vertex which is at distance 3, say v, and add the path from H_0 to v into H_i to form H_{i+1}. After t steps, we will have $1 + 3t$ vertices in H_t. Let a_t be the set of vertices at a distance at most 1 from H_t. Since H_0 is a single vertex and G has a minimum degree k, $a_0 \geq k+1$. For H_t, we have $a_t \geq a_{t-1} + k + 1$. a_t can never exceed the order of G, so we get $t \leq \lfloor \frac{n}{k+1} \rfloor - 1$. Therefore, H can have at most $3\lfloor \frac{n}{k+1} \rfloor - 2$ vertices. $\square$

We can apply the Theorem 32 to identify a subtree such that every vertex in the original graph is within a distance of at most 2 from some vertex in this subtree. By utilizing theorem 27, we can effectively burn the subtree and then allow the fire to spread for an additional 2 steps to ensure that the entire graph is completely burned.

Theorem 33. *(Martinsson, 2023, [20]) For any graph G with n vertices and minimum degree k, $b(G) \leq (1 + o(1))\sqrt{\frac{3n}{k+1}}$.*

Proof. For the given graph G, we can find a subgraph H such that all vertices of G are within distance at most 2 from H using the method described in Theorem

22. Since the burning number conjecture holds asymptotically, we can burn H in $(1+o(1))\lceil\sqrt{\frac{3n}{k+1}}\rceil - 2$ steps. Allowing H to burn for 2 more steps ensures that all vertices of G are burned. Thus, $b(G) \leq (1+o(1))\lceil\sqrt{\frac{3n}{k+1}}\rceil$. □

3.3.4 Directed Graphs of Janssen [18]

The upper bound of the burning number for single-source Directed Acyclic Graphs (DAGs) was given by Remie Janssen.

Theorem 34. *(Janssen, 2020, [18]) The burning number of a single-source Directed Acyclic Graph (DAG) with n vertices is at most $\lceil\sqrt{2n+\frac{1}{4}}-\frac{1}{2}\rceil$.*

Proof. We sketch the proof of [18]. Let k be the burning number of the DAG. With burning sources of radii $\{0,1,\ldots,k-1\}$, we can burn vertices in a range that allows us to cover at least $\{1,2,\ldots,k\}$ vertices. Thus, the total number of vertices that can be burned must be at least n, the order of the graph. Therefore:

$$\sum_{i=1}^{k} i \geq n$$

$$\frac{k(k+1)}{2} \geq n$$

$$k^2 + k - 2n \geq 0$$

Solving this quadratic inequality for k, we get:

$$k \leq \lceil\sqrt{2n+\frac{1}{4}}-\frac{1}{2}\rceil$$

□

3.3.5 Spider Graph of Das et al. [9]

The conjecture 1 was proven for spiders by Das et al. A spider of order n has one vertex of degree at least 3 (called the head) and $n-1$ vertices of degree at most 2.

Theorem 35. *(Das et al., 2018, [9]) For a spider S of order n, $b(S) \leq \lceil \sqrt{n} \rceil$.*

Proof. Here, we sketch the proof of [9]. We will prove this theorem using induction. Consider the base case where $n = 4$. The spider with 4 vertices consists of a single vertex connected to 3 others. We can burn this spider in 2 steps by burning the head vertex.

Assume that for all spiders of order n^2, the burning number $b(S) \leq n$. We want to determine the burning number for spiders of order $(n+1)^2$.

Assume the spider has k paths attached to the head, and let m denote the number of paths with length at least n. Let P_i be the length of path i as depicted in Figure 3.3.2. We have $m \leq n$; otherwise, the total number of vertices would exceed $(n+1)^2$.

1. If $P_m \geq 2n+1$, we can burn a vertex in the middle of the path, which will burn at least $2n+1$ vertices. The remaining vertices, totaling $(n+1)^2 - (2n+1) = n^2$, can be burned in n steps by the induction hypothesis [9].

2. If $P_m \leq 2n$ and $m < n$, burn the head of the spider in the first step. All paths of length at most n will be burned, and the remaining paths will have a length at most n. The number of remaining vertices will be at most $n^2 - (m-1)^2$, which can be covered with n burning sources [9].

3. If $P_m \leq 2n$ and $m = n$, all paths of length at least n will have order at least $n(n+1)$. The remaining paths will have a total order of at most $n+1$. Burning the head in the first step will cover at least $n^2 + 1$ vertices. The remaining vertices can be burned in n steps [9].

4. If no path has length $\leq n$, it implies $k = m = n$. Burning the head in the first step will burn at least $n^2 - n + 3$ vertices. The remaining vertices will induce a path-forest with at most $3n - 2$ vertices, which can be burned in n steps.

Hence, in all cases, we can burn $(n+1)^2$ vertices in at most $n+1$ steps, proving the theorem. □

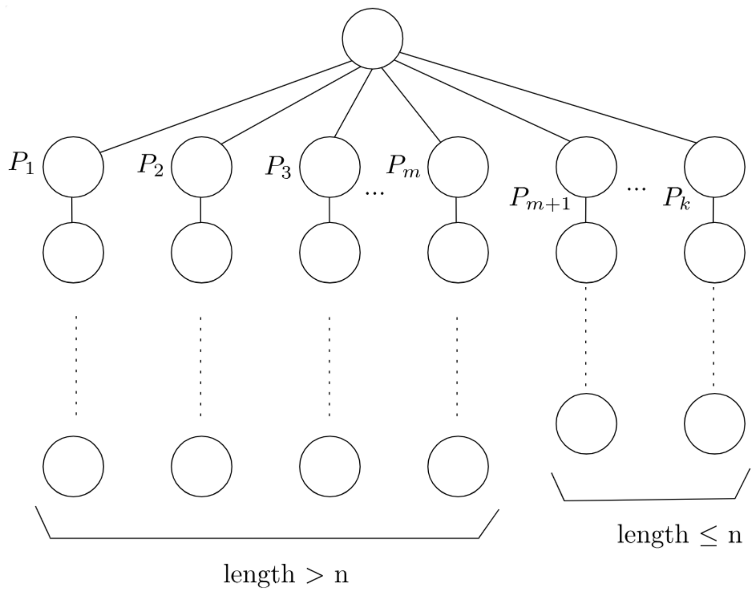

Fig. 3.3.2: Spider Graph Notations

CHAPTER 4

Firefighter Problem

In 1995, Hartnell [17] introduced the firefighter problem. In this problem, the fire starts at certain vertices in a graph, and firefighters aim to protect a set of vertices. During each step, the fire spreads to adjacent unprotected vertices, while firefighters work to protect new unburned vertices. The goal is to save the maximum number of vertices possible.

Firefighter Problem: Given an undirected graph $G = (V, E)$, the fire starts at vertex set $S \subset V$ and f firefighters are placed to protect f vertices which are not on fire. The protected vertices and burned vertices stay in the same state during the rest of the game. In the next step, the fire spreads to the adjacent unprotected vertices of all burned vertices and then f firefighters protect another set of vertices. The game continues until the fire can not spread to any more vertices. This problem is NP-complete [12].

For finite graphs, the main objective is to protect the maximum number of vertices. In the case of an infinite grid, the goal is to decide if f firefighters are sufficient to contain the fire. Containing the fire implies stopping it from spreading after a finite number of steps. For different variants, refer to the survey on firefighting [13].

In 2022, Burgess et al. [7] introduced the concept of distance-based restrictions within the context of the firefighter problem. In real-life scenarios, firefighters can travel a limited distance in a given time. That motivated them to propose that firefighters be restricted to moving a maximum distance of d in each step.

In 2023, Burgess et al. [6] further extended the distance restriction concept to finite graphs. They introduced two decision problems related to the firefighter problem, thereby broadening the scope of theoretical investigation in this area.

DR-b-FF

Input: An undirected graph $G = (V, E)$, the set of fire sources $S \subset V$, an integer distance $d \geq 1$, and a natural number $k \leq |V|$.

Question: Given that the fire breaks out on all vertices of S in graph G, can b firefighters protect at least k vertices if they are allowed to move a maximum distance of d?

DPR-b-FF

Input: An undirected graph $G = (V, E)$, the set of fire sources $S \subset V$, an integer distance $d \geq 1$, and a natural number $k \leq |V|$.

Question: Given that the fire breaks out on all vertices of S in graph G, can b firefighters protect at least k vertices if they are allowed to move a maximum distance of d without passing through any burning vertex?

Both problems have been proven to be NP-hard [6].

4.1 Current Results

For an infinite square grid, the number of firefighters required to contain the fire is well-studied.

Theorem 36. *(Days-Merrill, 2019, [11]) If firefighters can only move to adjacent vertices ($d = 1$), then three firefighters are insufficient to contain the fire in a square grid.*

Proof. Here, we sketch the proof of [11]. To contain the fire, at least one vertex on

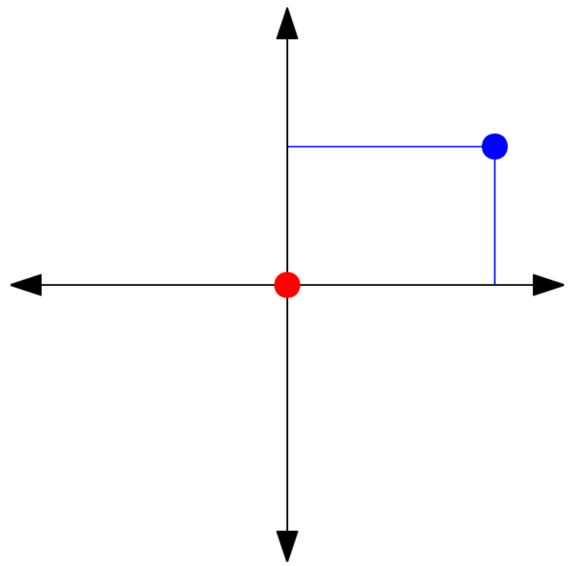

Fig. 4.1.1: One firefighter can save at most one axis

each axis must be protected. However, each firefighter can protect only one axis. Suppose a firefighter starts at a vertex $(0, y)$ (without loss of generality) and moves towards the positive x-axis to protect a vertex. The firefighter needs to travel a distance of at least $x + y$ to protect the vertex $(x, 0)$, but the fire will reach $(x, 0)$ in just x steps, meaning the vertex will burn before the firefighter can arrive to protect it.

If the firefighter instead starts at (x, y) where $x \geq y$, they may be able to protect a vertex on the positive x-axis. However, after protecting that vertex, the firefighter will need more time to reach any vertex on the positive y-axis, as both the fire and the firefighter move at the same speed. Consequently, a single firefighter can protect at most one axis.

Thus, with three firefighters, it is impossible to protect all necessary axes in the square grid, and the fire cannot be contained when $d = 1$. $\square$

Similarly, in the strong grid, we need to deploy 4 firefighters to protect the four principal axes. However, due to the faster spread of the fire in the strong grid, additional firefighters are required to safeguard the vertices along the lines at a 45-degree angle, specifically those on the lines $x = y$ and $x = -y$.

Theorem 37. *(Days-Merrill, 2019, [11]) In the strong grid, seven firefighters are insufficient to contain the fire if they are restricted to moving to adjacent vertices.*

Proof. We sketch the proof of [11]. In the strong grid, to contain the fire, at least one vertex on each of the four axes (positive and negative x- and y-axes) must be protected, along with two points on the line ($x = y$) and two points on the line ($x = -y$). This requires eight points in total, meaning we need one firefighter for each point.

Suppose a firefighter starts at a point on the positive y-axis, say $(0, y)$, and moves toward a point on the line $x = y$, such as (x', x'), to protect it in the first quadrant. To successfully contain the fire, the firefighter must move either horizontally or vertically, as moving diagonally would leave a gap for the fire to spread. Let (x', x') be the closest point on the line $x = y$, which would be (y, y).

Both the fire and the firefighter require y steps to reach this point, but since the fire starts spreading first, the vertex (y, y) will burn before the firefighter can protect it. Similarly, if the firefighter starts on the line $x = y$, they won't be able to protect any vertex on the axes. Even if the firefighter begins somewhere in between and tries to protect either a line or an axis, it will take them longer to reach the next critical point than the fire.

Consequently, each firefighter can protect at most one axis or line. Since we need to protect eight points in total, seven firefighters are not enough to contain the fire. □

When the movable distance for firefighters is increased to 2 units, we can successfully contain the fire in the strong grid using just 4 firefighters.

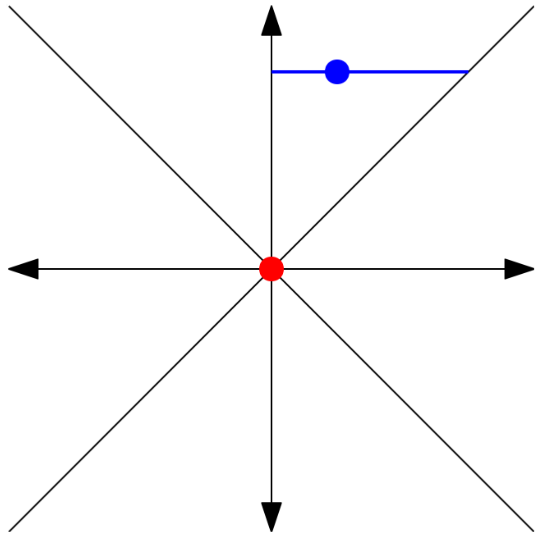

Fig. 4.1.2: One firefighter can save at most one point on either axis or on the line $x = y$ or $x = -y$

Theorem 38. *(Burgess et al., 2022, [7]) (Messinger, 2005, [22]) For $d \geq 2$, at least four firefighters are required to contain the fire on the strong grid.*

Proof. We sketch the proof of [7, 22]. Burgess, Marcoux, and Pike demonstrated that four firefighters are sufficient to contain the fire in the strong grid when $d = 2$ [7]. Figure 4.1.3 illustrates the fire's origin (represented as a red square) and the initial positions of the four firefighters (represented as blue squares).

Margaret-Ellen Messinger further proved that four firefighters are necessary, even without distance restrictions [22]. Imagine a fire starts at the point (0, 0) on the grid, and three firefighters immediately try to protect the nearby vertices. The goal of the firefighters is to "surround" the fire by creating walls of protected vertices around it. These walls would stop the fire from spreading any further. The idea is that at some time $t = k + j$ where $k, j > 0$, the firefighters would have built all three walls to contain the fire. However, there's a problem. By time $t = k + j$, the firefighters would need to protect more vertices than they can actually handle. The math shows that the firefighters can protect fewer vertices than the number of vertices needed to build the walls. Specifically, the number of vertices that need protection is too high

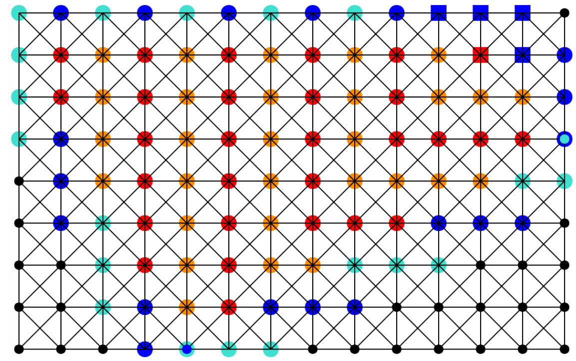

Fig. 4.1.3: Four firefighters are sufficient to contain the fire in the infinite strong grid for $d = 2$.

compared to what the three firefighters can protect at the same time. The detailed proof is given in [22].

□

In the square grid, it is possible to contain the fire with just 3 firefighters when the distance parameter d is set to 2.

In the following figures, the initial position of fire is marked as a red square and the vertices burned in each step are shown in red and orange disks alternatively. The initial firefighter positions are blue squares and their positions in different time are shown in cyan and blue disks in alternate manner.

Theorem 39. *(Burgess et al., 2022, [7]) In the infinite square grid, a single source of fire can be contained using three firefighters when $d = 2$.*

Proof. In the square grid, three firefighters are sufficient to contain the fire when they are allowed to move up to two vertices per step ($d = 2$). Initially, the three firefighters protect three adjacent vertices around the fire's origin. As the fire spreads in the remaining direction, two of the firefighters can move parallel to the fire's spread, protecting adjacent vertices as they go. Meanwhile, the third firefighter can utilize the increased movement distance to jump ahead by two vertices and cover the fire from the remaining direction, as shown in Figure 4.1.4. □

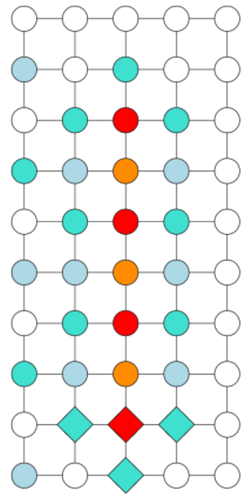

Fig. 4.1.4: Three firefighters are sufficient to contain the fire in the infinite square grid for $d = 2$

For $d = 2$ in the square grid, Burgess et al. [7] conjectured that two firefighters are insufficient to contain the fire.

Conjecture 40. *In the square grid, it is not possible to contain the fire with two firefighters if they are restricted to moving up to two vertices per step* $(d = 2)$*. [7]*

For distances greater than 2, we can always contain the fire in a square grid using just 2 firefighters. The same result can be applied when distance restriction is not applied.

Theorem 41. *(Burgess et al., 2022, [7]) The fire in the square grid can be contained with two firefighters when* $d \geq 3$*.*

Proof. As demonstrated by Burgess, Marcoux, and Pike, consider a fire that starts at the origin $(0, 0)$ in the square grid [7]. Starting with two firefighters, they can contain the fire by spiraling outward from the origin, as depicted in Figure 4.1.5, utilizing their ability to move up to three vertices per step $d \geq 3$. □

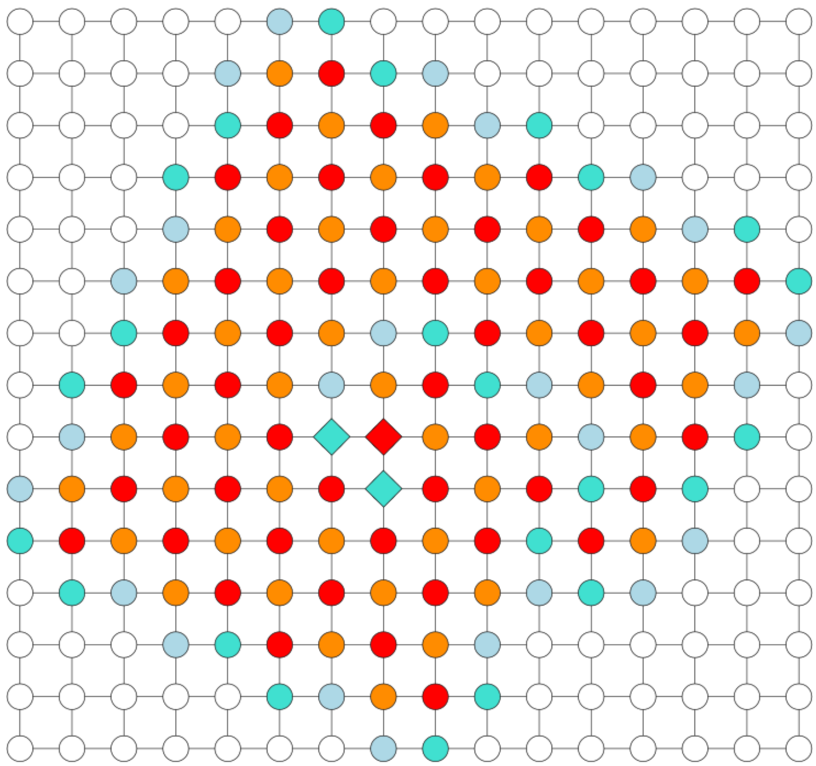

Fig. 4.1.5: Two firefighters are sufficient to contain the fire in the infinite square grid for $d \geq 3$

CHAPTER 5

Conclusion and Future Work

In this thesis, we have analysed two critical problems arising from the spread of information: the graph burning problem and the firefighter problem.

The graph burning is a discrete-time process that models the spread of information in the network. Since the problem has been proven to be NP-complete, we have summarised much of the literature that proposed the mathematical bounds and approximation algorithms. We have highlighted the continuous advancements in the field that led to the current state-of-the-art results in order to prove the famous burning number conjecture.

In contrast, the firefighter problem deals with the challenge of minimising the spread of falsified information or rumours. In our thesis, we mostly focused on distance-restricted firefighting where the firefighters have additional constraints.

Overall, the thesis summarised the main ideas and important advancements in the field of the mathematical study of information spreading in terms of graph burning and firefighter problems. The survey aims to provide a comprehensive study for future research.

5.1 Future Works

Future research can include investigating other restrictions on the firefighters by varying the distance allowed for them to move in each step. We can also analyse the results for increased number of firefighters.

Additionally, the conjectures 1 and 2 are still open problems which can be studied further.

Moreover, investigating the problem for different types of grids like strong grids or hexagonal grids can give us more insights. We can also change the properties of fire spreading to add further complexity to the problem which can give us a better understanding of the problem characteristics.

REFERENCES

[1] Bastide, P., Bonamy, M., Bonato, A., Charbit, P., Kamali, S., Pierron, T., and Rabie, M. (2023). Improved pyrotechnics: Closer to the burning number conjecture. *Electron. J. Comb.*, 30(4).

[2] Bessy, S., Bonato, A., Janssen, J., Rautenbach, D., and Roshanbin, E. (2017). Burning a graph is hard. *Discrete Applied Mathematics*, 232:73–87.

[3] Bessy, S., Bonato, A., Janssen, J., Rautenbach, D., and Roshanbin, E. (2018). Bounds on the burning number. *Discrete Applied Mathematics*, 235:16–22.

[4] Bonato, A., Janssen, J., and Roshanbin, E. (2014). Burning a graph as a model of social contagion. In *Algorithms and Models for the Web Graph: 11th International Workshop, WAW 2014, Beijing, China, December 17-18, 2014, Proceedings 11*, pages 13–22. Springer.

[5] Bonato, A. and Kamali, S. (2019). Approximation algorithms for graph burning. In *Theory and Applications of Models of Computation: 15th Annual Conference, TAMC 2019, Kitakyushu, Japan, April 13–16, 2019, Proceedings 15*, pages 74–92. Springer.

[6] Burgess, A. C., Hawkin, J., Howse, A., Marcoux, J., and Pike, D. A. (2023). Distance-restricted firefighting on finite graphs. *arXiv preprint arXiv:2306.12575*.

[7] Burgess, A. C., Marcoux, J., and Pike, D. A. (2022). Firefighting with a distance-based restriction. *CoRR*, abs/2204.01908.

[8] Clark, B. N., Colbourn, C. J., and Johnson, D. S. (1990). Unit disk graphs. *Discrete mathematics*, 86(1-3):165–177.

[9] Das, S., Dev, S. R., Sadhukhan, A., Sahoo, U. K., and Sen, S. (2018). Burning spiders. In *Conference on Algorithms and Discrete Applied Mathematics*, pages 155–163. Springer.

[10] Das, S., Islam, S. S., Mitra, R. M., and Paul, S. (2023). Burning a binary tree and its generalization. *CoRR*, abs/2308.02825.

[11] Days-Merrill, S. (2019). *Firefighter problem played on infinite graphs*. PhD thesis, Honors thesis, Bridgewater State University.

[12] Finbow, S., King, A., MacGillivray, G., and Rizzi, R. (2007). The firefighter problem for graphs of maximum degree three. *Discrete Mathematics*, 307(16):2094–2105. EuroComb '03 - Graphs and Algorithms.

[13] Finbow, S. and MacGillivray, G. (2009). The firefighter problem: a survey of results, directions and questions. *Australas. J Comb.*, 43:57–78.

[14] García-Díaz, J., Pérez-Sansalvador, J. C., Rodríguez-Henríquez, L. M. X., and Cornejo-Acosta, J. A. (2022). Burning graphs through farthest-first traversal. *IEEE Access*, 10:30395–30404.

[15] Gautam, R. K., Kare, A. S., and Bhavani, S. D. (2023). Approximation algorithms for the graph burning on cactus and directed trees. *arXiv preprint arXiv:2307.08505*.

[16] Gokhale, P., Keil, J. M., and Mondal, D. (2023). Improved and generalized algorithms for burning a planar point set. In *International Conference and Workshops on Algorithms and Computation*, pages 90–101. Springer.

[17] Hartnell, B. (1995). Firefighter! an application of domination. In *the 24th Manitoba Conference on Combinatorial Mathematics and Computing, University of Minitoba, Winnipeg, Cadada, 1995.*

[18] Janssen, R. (2020). The burning number of directed graphs: Bounds and computational complexity. *CoRR*, abs/2001.03381.

[19] Keil, J. M., Mondal, D., and Moradi, E. (2022). Burning number for the points in the plane. *CoRR*, abs/2205.04643.

[20] Martinsson, A. (2023a). Finding a good tree to burn. *arXiv preprint arXiv:2303.14039*.

[21] Martinsson, A. (2023b). On the approximability of the burning number. *CoRR*, abs/2308.04390.

[22] Messinger, M.-E. (2005). *Firefighting on infinite grids*. Master's thesis, Department of Mathematics and Statistics, Dalhousie University.

[23] Murakami, Y. (2024). The burning number conjecture is true for trees without degree-2 vertices. *Graphs Comb.*, 40(4):82.

[24] Nieberg, T. and Hurink, J. (2006). A ptas for the minimum dominating set problem in unit disk graphs. In Erlebach, T. and Persinao, G., editors, *Approximation and Online Algorithms*, pages 296–306, Berlin, Heidelberg. Springer Berlin Heidelberg.

[25] Norin, S. and Turcotte, J. (2024). The burning number conjecture holds asymptotically. *Journal of Combinatorial Theory, Series B*, 168:208–235.

[26] Shabanijou, S. K. M. (2023). Improved algorithms for burning planar point sets. In Pankratov, D., editor, *Proceedings of the 35th Canadian Conference on Computational Geometry (CCCG 2023) Montreal, Canada Denis Pankratov, Concordia University July 31 - August 4, 2023*, pages 161–167.

VITA AUCTORIS

NAME:	Jilsa Chandarana
PLACE OF BIRTH:	Ahmedabad, India
YEAR OF BIRTH:	2000
EDUCATION:	Charotar University of Science & Technology, Changa, Gujarat, 2018
	University of Windsor, M.Sc in Computer Science, Windsor, Ontario, 2020