The Firefighter Problem on Grids: Resistant Networks

A Senior Project submitted to
The Division of Science, Mathematics, and Computing
of
Bard College

by
Ha Phuong (Fiona) Do Thi

Annandale-on-Hudson, New York
May, 2013

Abstract

The firefighter problem models dynamic spread of fire on graphs, and the objective is to contain the spread with firefighters. This project investigates spread on rectangular grids with resistance levels. Resistant nodes delay the spread of fire, as it takes more time for a fire to burn a vertex and then further spread on other vertices. We determine the number of firefighters sufficient to contain a fire spread, and with our containment strategy we obtain the time the entire process takes. Our results build on previous studies of the same problem but dealing with graphs whose nodes do not have any resistance. We use a modeling environment NetLogo to simulate the spread and test different containment strategies.

Contents

List of Figures

Dedication

I dedicate this project to my brother, whose love and support have been crucial in every stage of my life.

Acknowledgments

First, I would like to thank my senior project advisor Sam Hsiao for his help, patience and guidance in every stage of this project and my computer science professor Sven Anderson for helping me learn NetLogo in such a short amount of time. I would also like thank professor Jim Belk for his useful comments and suggestions that helped me complete my project. Finally, I would like to thank all my family and friends who have been there for me, and without whom I would not survive all the sleepless nights completing this project.

1
Background

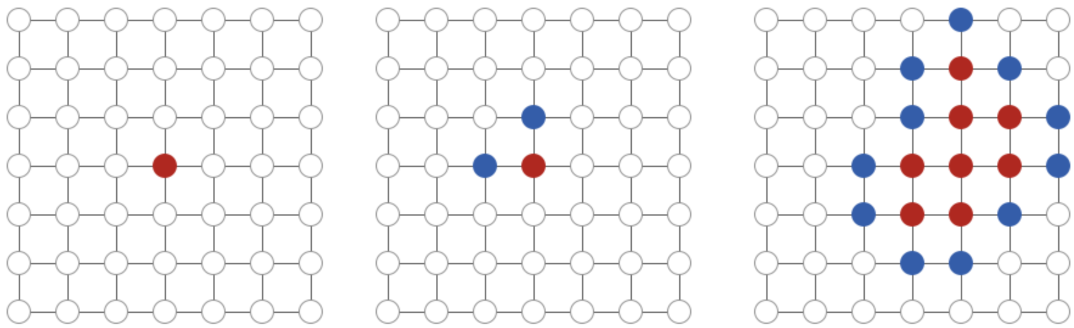
1.1 Introduction

The firefighter problem was first introduced by Bert Hartnell [5] in 1995 and it is a problem of dynamic spread of fire on graphs. The original case can be described as follows.

Let (G, V, r) be a connected rooted graph with $r \in V(G)$, where $V(G)$ is a set of vertices of G. A fire initially breaks out at vertex r. At the beginning of each subsequent discrete time interval, the fire from *burned* vertices spreads to all neighboring vertices that are not *protected* by firefighters. Then, at the end of the time interval, a number of firefighters are available to be positioned at vertices that are currently not on fire. Positioned firefighter remains on assigned vertex and *protects* it for the rest of the entire process. The fire is *contained* when it can no longer spread. Figure **??** shows an example of the whole process.

Among most often pursued objectives in this problem are:

- minimizing the number of burned vertices,
- determining the least number of firefighters necessary to contain a fire,
- (in finite graphs) maximizing the number of saved vertices,



(a) A fire breaks out (b) Firefighters are deployed (c) The fire is contained

Figure 1.1.1: Illustration of the firefighter problem on the grid.

- determining the minimal amount of time steps it takes to contain a fire.

These objectives are often reviewed separately, as they are sometimes incompatible.

This project explores a variation of the firefighter problem, in which a concept of *fire resistance* of vertices is introduced. Fire resistant nodes delay the spread and it takes more time intervals for a fire to *fully burn* a vertex. This approach examines effectiveness of networks with inherently stronger resistance to fire.

We show that one firefighter is sufficient to contain a fire spread on a quarter plane of resistant network. We provide an algorithm for containment, which yields the time it takes to contain the spread. Given the strategy, we also provide a general formula for containment and resulting time given any resistance level.

The framework of the problem is introduced in Chapter 2. Definitions, terminology, and graphs are provided. The problem is first examined on a quarter plane in Chapter 3. These explanations and results serve as the foundation to understanding proofs on the full plane. Investigation of the full plane is presented in Chapter 4. Further possible research is discussed in Chapter 5.

1.2 Prior Results

Various papers investigating different variations of the firefighter problem have appeared in literature.

Wang and Moeller [6] proved that two firefighters per time interval are sufficient to contain a single-source fire spread in an infinite rectangular grid.

Hartke [4] further proved that it takes minimum of eight time intervals to contain a fire with two firefighters. In this process, eighteen vertices are burned.

Fogarty [3] proved that in two-dimensional rectangular grid, any finite-source fire outbreak can be contained with two firefighters.

Finbow, Hartnell et al [2]. also researched viruses on a network with the aim of minimizing the damage incurred. They focused on constructing optical graphs to minimize the number of infected vertices given that all graphs share the same number of vertices.

Hartnell [5] originally proposed the problem of firefighting in terms of resistance movements. He wanted to know how to establish an underground communications network which minimizes the effects of treachery by a member of the network which is followed by the betrayal of the other members

2
Preliminaries

This chapter introduces definitions, notations, and graphs that thoroughly explain the framework of the investigated problem. Some of the definitions and terms can be found analogous to those of Fogarty [3] and Feldheim [1]. Presented terminology provides foundation for analysis of resistant networks.

2.1 Definitions

Graphs

The terminology for graphs follows the notation presented in [7]. Let $G = (V, E)$ be a graph, so G consists of a vertex (or node) set $V(G)$ and an edge set $E(G)$. The edge set $E(G)$ is a set of unordered pair of vertices $\{v, w\}$ for $v, w \in V(G)$. Suppose for $e \in E(G)$ we have $e = \{v, w\}$, then v and w are adjacent. This project investigates the firefighter problem on an the infinite Cartesian grid, and so we let the set of vertices be $V(G) = \mathbb{Z}\times\mathbb{Z}$ and edge set be $E(G) = \{((x_1, y_1), (x_2, y_2)) : x_1.x_2, y_1, y_2 \in \mathbb{Z} \text{ and } |x_1-x_2|+|y_1-y_2| = 1\}$.

Firefighters

Firefighters can be placed on vertices that are not on fire. A firefighter *protects* a vertex when it is placed on a vertex which is not yet on fire and does not already have a firefighter assigned. Once deployed, firefighters remain on their assigned positions and protect their nodes through the rest of the process. Fire cannot spread on protected vertices. The number of firefighters available for deployment at time t is denoted as $f(t)$, and let $F(t) = \sum_{i=1}^{t} f(i)$ be the sum of all firefighters deployed from time 1 to time t.

Fire spread

Fire spread can be referenced to as *unconstrained* or *contained*. *Unconstrained* fire spread means that there is no intervention of firefighters at the time, whereas if firefighters are deployed and the whole process terminates at a finite number of time intervals, the fire becomes *contained* and can no longer spread.

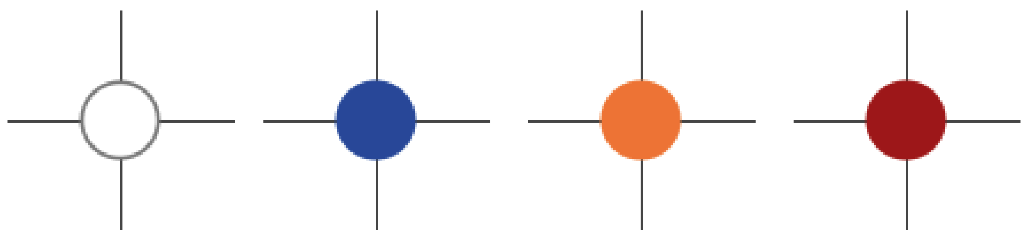
To avoid possible ambiguities, we define precise timeline for the process. Table 2.1.1 shows the timeline for the process of fire spread in the original problem, i.e. on networks with no resistance.

time	process that happened
beginning of time 1	
	- fire starts at the origin - firefighters are deployed
end of time 1	
beginning of time 2	
	- fire spreads to neighboring nodes - firefighters are deployed
end of time 2	
beginning of time n	
	- fire spreads to neighboring nodes - firefighters are deployed
end of time n	

Table 2.1.1: Timetable of the process of fire spreading and protecting the nodes.

Nodes

We define different possible states of vertices. A node can be *burning, burned, protected*, or *vulnerable*. A vertex that caught on fire is defined as *burning*. A *burning* vertex becomes *burned* if at the next time step, the fire would spread to its neighbors. The distinction between a *burning* and a *burned* vertex is important in this project, because in the original problem a node that caught on fire immediately becomes *burned*, but we investigate cases when a node is *burning* for 1 or more time intervals before it becomes *burned*. A *protected* vertex has an assigned firefighter and cannot catch on fire. *Vulnerable* nodes are not burning nor protected. A legend of different states of vertices is presented in Figure 2.1.1.



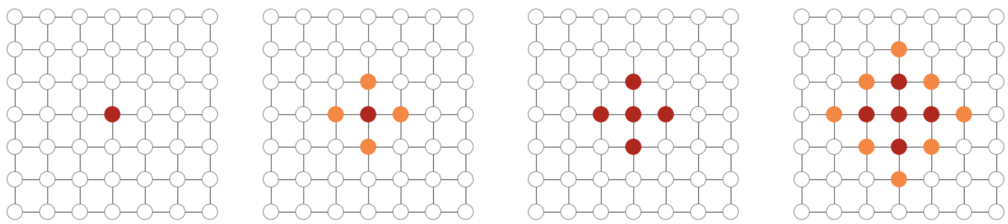
(a) Vulnerable node (b) Protected node (c) Burning node (d) Burned node

Figure 2.1.1: Legend of different states of nodes.

Resistance level

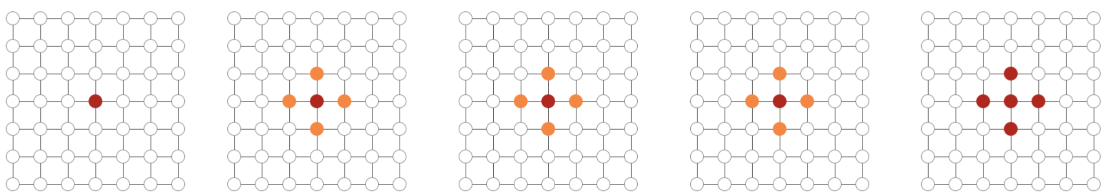
Resistance level of a node, denoted ω, is the number of time intervals it takes the node to change the state from *burning* to *burned*. Example 2.1.1 illustrates spread of fire on networks with resistance.

Example 2.1.1. Suppose a network has resistance 1. At time 1, a fire breaks out at the origin. Then at time 2, it spreads to adjacent vertices. Those nodes catch on fire and are said to be *burning* nodes. Nodes are level 1 resistant, so the fire burns for one time step. At time 3, the *burning* nodes become *burned*, and ready to spread fire at the next time step. At time 4, the fire burns the adjacent vertices. Figure 2.1.2 illustrate this effect on the graph. Figure 2.1.3 shows analogous spread on a network with resistance 3. ◊



(a) time 1 (b) time 2 (c) time 3 (d) time 4

Figure 2.1.2: Unconstrained fire spread on network with resistance 1.



(a) time 1 (b) time 2 (c) time 3 (d) time 4 (e) time 4

Figure 2.1.3: Unconstrained fire spread on network with resistance 3.

Potential and potential fronts

In our analysis, we utilize concepts of *potential* and *potential fronts*, which were introduced by Feldheim [1]. We define values $n_i(t)$'s as following:

$$n_1(t) = \max\{|x+y| : (x,y) \text{ is burning at time t }\} + 1$$

$$n_2(t) = \max\{|-x+y| : (x,y) \text{ is burning at time t }\} + 1$$

$$n_3(t) = \max\{|-x-y| : (x,y) \text{ is burning at time t }\} + 1$$

$$n_4(t) = \max\{|x-y| : (x,y) \text{ is burning at time t }\} + 1$$

Potential fronts presented in Figure 2.1.4 are diagonal lines $L_i(t)$'s defined as following:

$$L_1(t) = \{(x, y) : x + y = n_1\}$$

$$L_2(t) = \{(x, y) : -x + y = n_2\}$$

$$L_3(t) = \{(x, y) : -x - y = n_3\}$$

$$L_4(t) = \{(x, y) : x + y = n_4\}$$

Each potential front $L_i(t)$ binds the spread at time t in the direction of the corresponding quadrant i. The values $n_i(t)t$'s can be understood as by how much the potential front shifted from the origin at time t.

Potential at time t in the direction of quadrant i, denoted $P_i(t)$ is the number of *vulnerable* nodes on the corresponding potential front $L_i(t)$ that are adjacent to *burning* nodes. Note that if at time t all *burning* nodes are *burned*, then the potential denotes the number of vertices that catch on fire at time $t + 1$. We denote the total potential from all $L_i(t)$'s as $P(t) = \sum_{i=1}^{4} P_i(t)$.

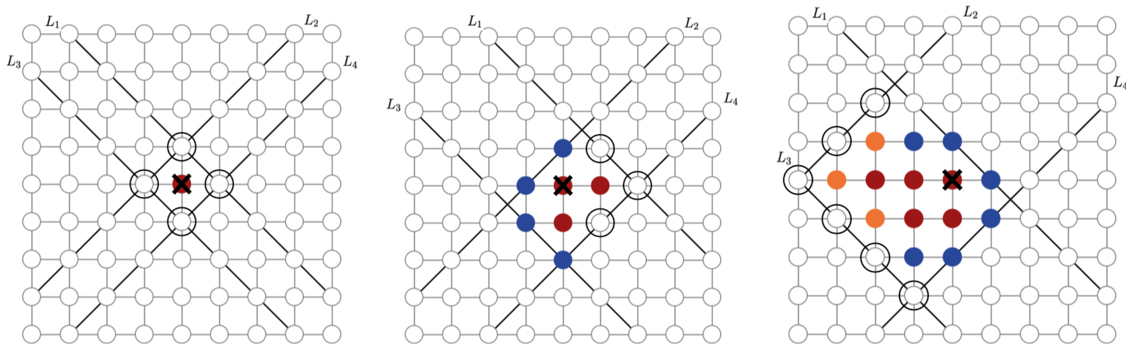
The examples in the next part will clarify these values and definitions.

Examples and Illustrations

We provide graphs and discussions to illustrate the concepts defined in this section. Figure 2.1.4 presents examples of different possible scenarios in the firefighter problem and outlined potential fronts and nodes that contribute to the potential. The origin of the fire outbreak is marked with the "x" and nodes that contribute to the potential are circled.

Note: Subscript i is added to firefighter values $f_i(t)$ and $F_i(t)$ to denote the values referring to firefighters that were placed on potential fronts $L_i(t)$. The notation without the subscript means the total values from all fronts, i.e. $f(t) = \sum_{i=1}^{4} f_i(t)$.

Note: For nodes that lay on intersection of two potential fronts, the value of firefighter or potential is counted as $\frac{1}{2}$ for each front $L_i(t)$.



(a) scenario 1 (b) scenario 2 (c) scenario 3

Figure 2.1.4: Graphs at time t for different possible scenarios in the firefighter problem.

- Figure 2.1.4a scenario 1

This is a graph of a full plane at time 1, when a fire breaks out. The potential fronts shifted 1 from the origin, so we have $n_1(1) = n_2(1) = n_3(1) = n_4(1) = 1$. The potential on each front is counted from two nodes that lay on intersections, so we have $P_1(1) = P_2(1) = P_3(1) = P_4(1) = \frac{1}{2} + \frac{1}{2} = 1$. The total potential is equal to $P(t) = 4$, which corresponds to the 4 circled nodes that are adjacent to the burning origin. There are no firefighters in this scenario, so we have $f(1) = F(1) = 0$.

- Figure 2.1.4b scenario 2

Here we have total number of firefighters $f(t) = 4$, and if we break down to each potential front, we get $f_1(t) = 0$, $f_2(t) = 2$, $f_3(t) = 1.5$, $f_4(t) = 0.5$, which add up properly. The potential we see given by $P_1(t) = P_2(t) = 1.5$ and $P_2(t)t = P_3(t) = 0$, which gives $P(t) = 3$, which are the 3 circled nodes. In this case, some of the fronts shifted more than the others, so we have $n_1(t) = n_3(t) = n_4(t) = 2$ and $n_2(t) = 1$.

- Figure 2.1.4c scenario 3

In this case, we have $f_1(t) = f_4(t) = 2$, which add up to total $f(t) = 4$. For the potential, we get $P_1(t) = 0$, $P_2(t) = 2.5$, $P_3(t) = \frac{1}{2} + 2 + \frac{1}{2} = 3$, $P_4(t) = 0.5$. Thus, $P(t) = 6$. The potential fronts shifted by the following: $n_1(t) = 1$, $n_2(t) = n_3(t) = 4$, $n_4(t) = 2$.

2.2 Summary of Notation

We provide an overall summary of discussed notations and concepts. The brief form could be used as a helpful glossary of terms that appear throughout the whole project.

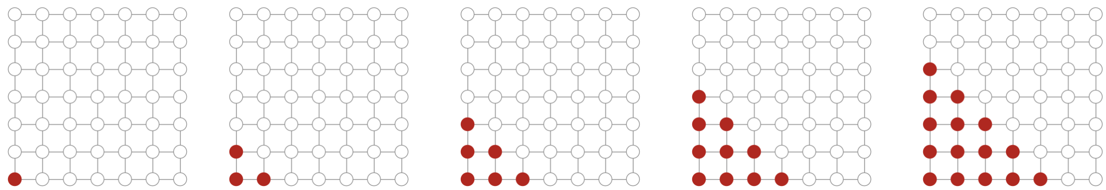
- Nodes can be *burning*, *burned*, *protected* or *vulnerable*,
- $L_i(t)$ denotes a potential front at time t that binds the spread in the direction of quadrant i.
- $n_i(t)$ denotes by how much the potential front L_i shifted from the origin at time t
- $f_i(t)$ denotes a number of firefighters deployed at time t on $L_i(t)$,
- $f(t) = \sum_{i=1}^{4} f_i(t)$
- $F_i(t) = \sum_{k=1}^{t} f_i(k)$ denotes the total number of firefighters deployed on $L_i(t)$ from the beginning to time t,
- $F(t) = \sum_{i=1}^{4} F_i(t)$
- $P_i(t)$ denotes the potential at time t on $L_i(t)$.
- $P(t) = \sum_{i=1}^{4} P_i(t)$
- ω denotes a resistance level of a network or a node.

3
Quarter Plane

3.1 Introduction

This section investigates fire spread on a quarter plane of resistant networks. Let G be a grid on a quarter plane, so we get $V(G) = \mathbb{Z}_{\geq 0} \times \mathbb{Z}_{\geq 0}$, where $\mathbb{Z}_{\geq 0}$ denotes the set of nonnegative integers. Note that we do not use subscript notation given we deal only with one quadrant. Figure 3.1.1 shows an unconstrained fire spread on a network with no resistance. Fogarty [3] proved that one firefighter is insufficient to contain such spread. In this section, we first look at a network with resistance 1, and we show that one firefighter deployed at any time after a fire breakout, is sufficient to contain the fire. We provide a strategy for containment and a resulting time. Finally, given the strategy, we are able to determine a general formula for the time of the entire process on a network with any resistance level.

As presented in the Figure 3.1.1, on a quarter plane, the fire on burned nodes spreads upwards and to the left. We shall prove the Claim 3.1.1 on the number of nodes on a potential front at time t.



(a) time 1 (b) time 2 (c) time 3 (d) time 4 (e) time 4

Figure 3.1.1: Unconstrained fire spread on a quarter plane with no resistance.

Claim 3.1.1. *Let G be the quarter plane. At time t there are $n(t)+1$ nodes on the potential front $L(t)$.*

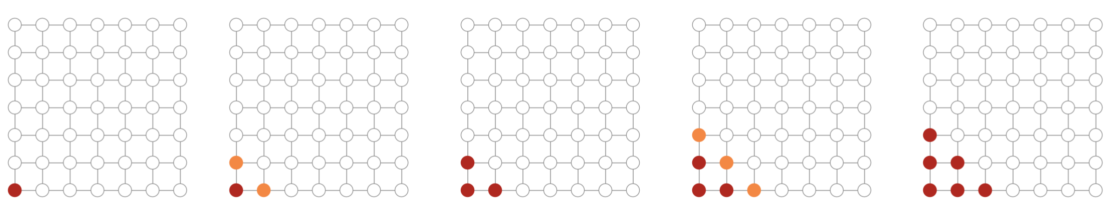
Proof. We know by definition of a potential front that all nodes $(x, y) \in L(t)$ satisfy $x + y = n(t)$ for $x, y \in \mathbb{Z}_{\geq 0}$. Given x, y are non-negative and $y = n(t) - x$, we get that $x, y \in [0, n(t)]$. We can conclude that all nodes on the potential front in a quarter plane can be given by

$$L(t) = \{(i, n(t) - i) : i \in \mathbb{Z}_{\geq 0} \text{ and } i \in [0, n(t)]\}.$$

Thus we have $\sum_{i=0}^{n(t)} 1 = n(t) + 1$ nodes on $L(t)$. □

3.2 Resistance Level 1

We begin with looking at a network with resistance level 1. Figure 3.2.1 shows unconstrained fire spread on the network. In this case, we can prove that one firefighter is sufficient to contain the fire.



(a) time 1 (b) time 2 (c) time 3 (d) time 4 (e) time 4

Figure 3.2.1: Unconstrained fire spread on a quarter plane with resistance 1.

Consider values $n(t)$ for the graphs presented in Figure 3.2.1. The Claim 3.2.1 determines the conditions of $n(t)$ and Claim 3.2.2 specifies the values of $n(t)$. The claims provide the foundation to understanding the proof of the Theorem 3.2.5.

Claim 3.2.1. *If fire is not contained at time t, we have following cases*

$$n(t+1) = \begin{cases} n(t)+1 & \text{if } t \text{ is odd} \\ n(t) & \text{if } t \text{ is even.} \end{cases}$$

Proof. When fire breaks out at time 1, one node becomes burned. Then, at each next time step t, we observe two actions that repeatedly happen in consecutive order:

- *case 1:* if at time $t-1$ all nodes on fire were burned, then at time t, the fire spreads to adjacent nodes and stays there for one time step, which gives $n(t+1) = n(t)$.

- *case 2:* if at time $t-1$ fire spread and new nodes caught on fire, then at time t they become burned, and so ready to spread fire at time $t+1$, which implies $n(t+1) = n(t)+1$. At time 1 we have all nodes on fire burned, so we get that case 1 happens for t even and case 2 for t odd. □

Claim 3.2.2. *If fire is not contained at time t, then we obtain value of n by the following formula*

$$n(t) = \begin{cases} \frac{t+1}{2} & \text{if } t \text{ is odd} \\ \frac{t+2}{2} & \text{if } t \text{ is even.} \end{cases}$$

Proof. We can show this by induction on t. At time 1, node (0,0) is burned, so we have $n(1) = 0+0+1 = 1 = \frac{1+1}{2}$. By Claim 3.2.1 we get that at time 2, $n(2) = n(1)+1 = 2 = \frac{2+2}{2}$. At time 3, we get $n(3) = n(2) = 2 = \frac{3+1}{2}$.

Now, consider two cases for time $k \in \mathbb{N}$:

case 1: Let k be even, and suppose $n(k) = \frac{k+2}{2}$. Then, we get $n(k+1)$ odd and

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

case 2: Let k be odd, and suppose $n(k) = \frac{k+1}{2}$. Then, we get $n(k+1)$ even and

$$n(k+1) = n(k)+1 = \frac{k+1}{2} + 1 = \frac{k+3}{2} = \frac{(k+1)+2}{2}.$$

□

We provide two additional remarks.

Remark 3.2.3. If fire spreads at time t, then the number of nodes that just caught on fire is equal to $P(t-1)$. ◊

Remark 3.2.4. Let $b(t)$ be the number of burning nodes with $\max\{x+y\}$, then the are $b(t)+1$ nodes on the potential front that are adjacent to those burning nodes. ◊

Now, we can proceed to proving our Theorem 3.2.5. We present a strategy that contains the fire spread.

Theorem 3.2.5. *Let $G = \mathbb{Z}_{\geq 0} \times \mathbb{Z}_{\geq 0}$ be the quarter plane with resistance level 1. If a fire spreads unconstrainedly for k time intervals, then one firefighter is sufficient to contain the fire spread.*

Proof. Suppose a fire spreads unconstrainedly for k time intervals. We shall present a strategy for containment and show that with the strategy, the potential eventually decreases to 0, which implies that the fire can no longer spread.

Without loss of generality, we suppose k is even. Figure 3.2.2 shows the graph at time k, for some k even.

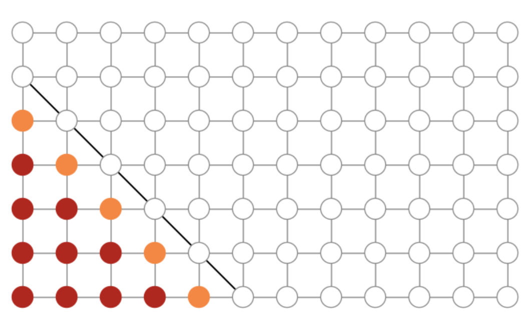


Figure 3.2.2: Graph of unconstrained fire spread at time k, for some k even.

We determine the initial values for $n(t)$ and potential as

$$n(k) = \frac{k+2}{2} \quad \text{and by Claim 3.1.1} \quad P(k) = n(k) + 1.$$

We introduce our first firefighter at $t = k+1$, thus we define our firefighter function by

$$f(t) = \begin{cases} 1 & \text{for } t > k \\ 0 & \text{for } t \leq k. \end{cases}$$

In our strategy, we place firefighters on potential fronts. By definition of $n(t)$, we know that at time t, nodes with coordinates $x + y = n(t)$ are not on fire.

We shall introduce our strategy. Define $\phi : \{k+1, k+2, ...\} \longrightarrow \mathbb{Z}_{\geq 0} \times \mathbb{Z}_{\geq 0}$, by

$$\phi(k+i) = \begin{cases} (i-1, n(k) - \frac{i-1}{2}) & \text{if } i \text{ is odd} \\ (i-1, n(k) - \frac{i-2}{2}) & \text{if } t \text{ is even,} \end{cases} \tag{3.2.1}$$

where $k, i \in \mathbb{N}$.

Consider $\phi(k+i)$ as the node, at which we deploy a firefighter at a given time, in order to contain a fire that spread unconstrainedly for k time intervals.

First, we make sure that every time firefighters are placed on non-burning nodes.

(1) Suppose i is odd. Then we get $\phi(k+i) = (i-1, n(k) - \frac{i-1}{2})$, and so

$$i - 1 + n(k) - \frac{i-1}{2} = i - 1 + \frac{k+2}{2} - \frac{i-1}{2} = \frac{(k+i)+1}{2} = n(k+i)$$

(2) Suppose i is even. Then we get $\phi(k+i) = (i-1, n(k) - \frac{i-2}{2})$, and so

$$i - 1 + n(k) - \frac{i-2}{2} = i - 1 + \frac{k+2}{2} - \frac{i-2}{2} = \frac{(k+i)+2}{2} = n(k+i).$$

In either case, $\phi(k+i)$ lies on the potential front $L(k+i)$, so the node is not burning.

We argue that given the strategy, we obtain the following formula for the potential values,

$$P(k+i) = \begin{cases} n(k) - \frac{i-1}{2} & \text{for } i \text{ odd} \\ n(k) - \frac{i-2}{2} & \text{for } i \text{ even,} \end{cases} \tag{3.2.2}$$

where $i \in \mathbb{N}$. We prove that Equation 3.2.2 is valid by induction on i.

First, we consider three first steps.

Let $i = 1$. At time $t = k+1$, which is odd, we have $n(k+1) = n(k)$. We place a firefighter on the fire front at $(0, n(k+1))$, which by Claim 3.2.1, we get $(0, n(k))$ as presented in Figure 3.2.3. We get the potential

$$P(k+1) = n(k+1) + 1 - f(k+1) = n(k) + 1 - 1 = n(k), \text{ and so}$$
$$P(k+1) = n(k) - \frac{1-1}{2}.$$

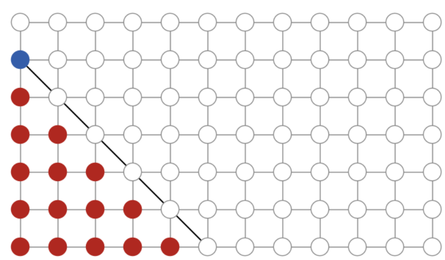


Figure 3.2.3: Graph at time $k+1$.

Let $i = 2$. At time $t = k+2$, which is even, the fire spreads and we have $n(k+2) = n(k+1) + 1 = n(k) + 1$. By Remark 3.2.3 we get that the number of nodes that just caught on fire is $P(k+1)$. We place a firefighter on the fire front at $(1, n(k))$ as presented in Figure 3.2.4. We get the potential

$$P(k+2) = P(k+1) + 1 - f(k+2) = n(k) + 1 - 1 = n(k), \text{ and so}$$
$$P(k+2) = n(k) - \frac{2-2}{2}.$$

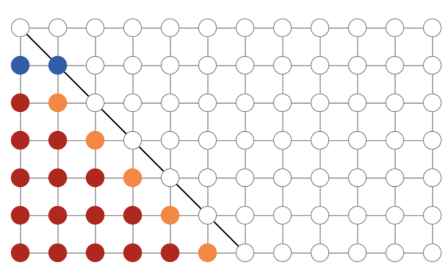


Figure 3.2.4: Graph at time $k+2$.

Let $i = 3$. At time $t = k+3$, which is odd, we have $n(k+3) = n(k+2) = n(k+1)+1 = n(k)+1$. We deploy a firefighter o n the fire front at $(2, (n(k)-1))$ as presented in Figure 3.2.5. We get the potential

$$P(k+3) = P(k+2) - f(k+2) = n(k) - 1, \text{ and so}$$
$$P(k+3) = n(k) - \frac{2-1}{2}.$$

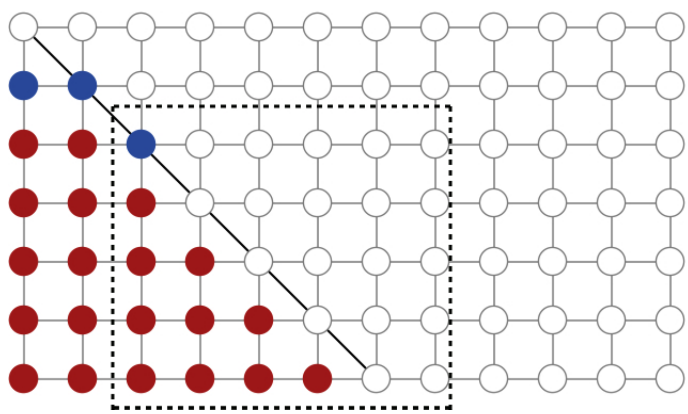


Figure 3.2.5: Graph at time $k+3$.

We observe that the graph at time $k+3$ is analogous to the one at time $k+1$, as outlined with the dotted line in Figure 3.2.5. At time $k+3$ we land back at the same graph structure, but at a reduced size. We get that

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

By induction, we know the strategy works on the dotted graph. Every two time steps we get an analogous graph with adjusted k equal to previous $k-2$. The nested graphs are presented in Figure 3.2.6

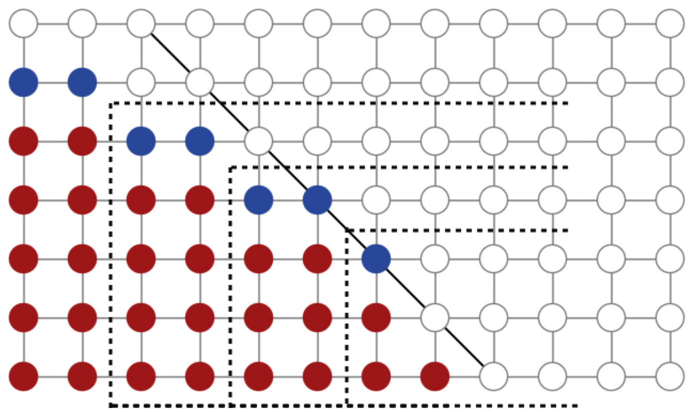


Figure 3.2.6: Illustration of nested graphs in G.

We identify that with this strategy, the potential satisfies two cases:

$P(t+1) = P(t)$ for t odd and $P(t+1) = P(t) - 1$ for t even.

Thus, continuing with induction, we check $i > 3$.

(1) Let i be odd, and suppose $P(k+i) = n(k) - \frac{i-1}{2}$. Then we get

$$P(k+i+1) = P(k+i) = n(k) - \frac{i-1}{2} = n(k) - \frac{(i+1)-2}{2}.$$

(2) Let i be even, and suppose $P(k+i) = n(k) - \frac{i-2}{2}$. Then we get

$$P(k+i+1) = P(k+i) - 1 = n(k) - \frac{i-2}{2} - 1 = n(k) - \frac{(i+1)-1}{2}.$$

We showed that the potential formula is valid. Therefore, given $P(t)$ is a decreasing every two time steps, we get that the spread will eventually be contained.

Additionally, we can determine when the fire is contained, which gives $P(t) = 0$. Since $P(t) = P(t+1)$ for t odd, we can focus on $P(t)$ for odd t and solve for i.

$$\begin{aligned} P(k+i) &= 0 \\ n(k) - \frac{i-1}{2} &= 0 \\ n(k) &= \frac{i-1}{2} \end{aligned}$$

we know that by Claim 3.2.2 that $n(k) = \frac{k+2}{2}$, so

$$\frac{k+2}{2} = \frac{i-1}{2}$$
$$i = k+3.$$

Therefore, we can conclude that for any k, one firefighter per time step is enough to contain a fire spread on a quarter plane with resistance level 1. Applying our strategy $\phi(t)$ for $t > k$, we determined that it would take $k+3$ time intervals, to complete the process. The graph of the completed process is presented in Figure 3.2.7.

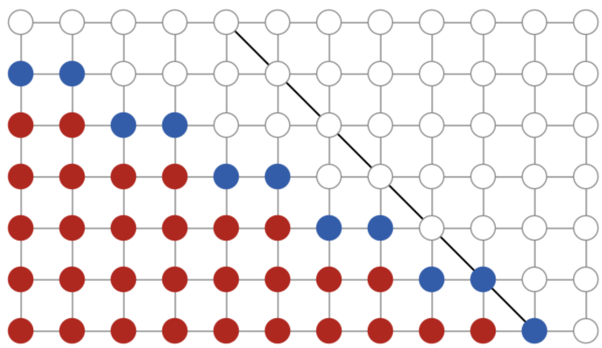


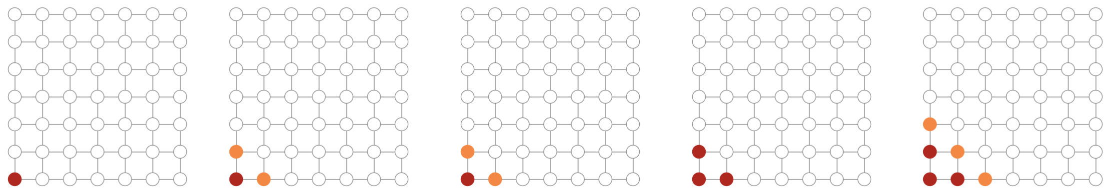
Figure 3.2.7: Contained fire spread at time $k+3$.

$\square$

3.3 Resistance level ω

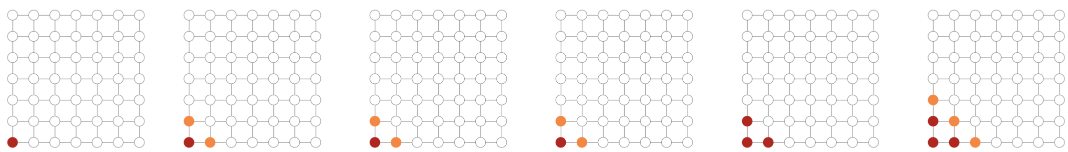
This section considers fire spread on a quarter plane with resistance level ω. Like we assumed in the previous section, we let a fire to spread unprotected for k time steps. We know that one firefighter will definitely be sufficient to contain a fire spread, but we will investigate how much time it takes to contain it.

First, we look at unconstrained spread of fire on the network. The Figure 3.3.1 and Figure 3.3.3 bellow show unconstrained spread on graphs with different resistance levels from time 1 until it spreads further to neighboring nodes.



(a) time 1 (b) time 2 (c) time 3 (d) time 4 (e) time 5

Figure 3.3.1: Unconstrained fire spread on a quarter plane with resistance 2.



(a) time 1 (b) time 2 (c) time 3 (d) time 4 (e) time 5 (f) time 6

Figure 3.3.2: Unconstrained fire spread on a quarter plane with resistance 3.

First, we shall consider values for $n(t)$ when the fire is not yet contained. We see that at time 1, the fire breaks out, and at time 2 it spreads. It takes ω time steps for the fire to fully burn the nodes, and so the nodes become burned at time $2+\omega$ and at time $2+\omega+1$ the fire spreads again. Thus, we can conclude with the following remark.

Remark 3.3.1. Let G be a quarter plane with resistance ω. Suppose at time t the fire is not contained, then the following conditions apply for value of $n(t)$

$$n(1) = 1 \tag{3.3.1}$$

$$n(2) = 2 \tag{3.3.2}$$

$$n(t+1) = \begin{cases} n(t)+1 & \text{for } t = 1 \mod (\omega+1) \\ n(t) & \text{fot } t = m \mod (\omega+1) \text{ for } m \in \mathbb{Z} \cap [0,\omega] - \{1\} \end{cases} \tag{3.3.3}$$

$\lozenge$

This means that when the fire burns the nodes, the potential front remains the same, and when it spreads, the potential shifts by one.

Claim 3.3.2. *Let G be a quarter plane with resistance ω. Suppose at time t the fire is not contained, then we get the following formula for value for $n(t)$*

$$n(t) = \begin{cases} \frac{t+2\omega}{\omega+1} & \textit{for } t = 2 \mod (\omega+1) \\ \frac{t+2\omega-[(t-2) \mod (\omega+1)]}{\omega+1} & \textit{fot } t = m \mod (\omega+1) \textit{ for } m \in \mathbb{Z} \cap [0,\omega] - \{2\} \end{cases}$$

Proof. We prove this claim by induction on t. First consider several cases for t.

Let $t = 1$. Then we get

$$\begin{aligned} n(1) &= \frac{1+2\omega-[(1-2) \mod (\omega+1)]}{\omega+1} \\ &= \frac{1+2\omega-[-1 \mod (\omega+1)]}{\omega+1} \\ &= \frac{1+2\omega-w}{\omega+1} \\ &= 1. \end{aligned}$$

Let $t = 2$, Then we get

$$\begin{aligned} n(2) &= \frac{2+2\omega}{\omega+1} \\ &= 2 \end{aligned}$$

Now, let time $k = 1 \mod (\omega+1)$, and suppose $n(k) = \frac{k+2\omega-[(k-2) \mod (\omega+1)]}{\omega+1}$. Then note that $k+1 = 2 \mod (\omega+1)$ and so by Remark 3.3.2 (3), we get

$$\begin{aligned} n(k+1) &= n(k)+1 \\ &= \frac{k+2\omega-[(k-2) \mod (\omega+1)]}{\omega+1}+1 \\ &= \frac{k+2\omega-[(1-2) \mod (\omega+1)]+\omega+1}{\omega+1} \\ &= \frac{t+2\omega-\omega+\omega+1}{\omega+1} \\ &= \frac{k+1+2\omega}{\omega+1}. \end{aligned}$$

Now, let time $k = 2 \mod (\omega+1)$, and suppose $n(k) = \frac{k+2\omega}{\omega+1}$. Then by Remark 3.3.2 (3), we get

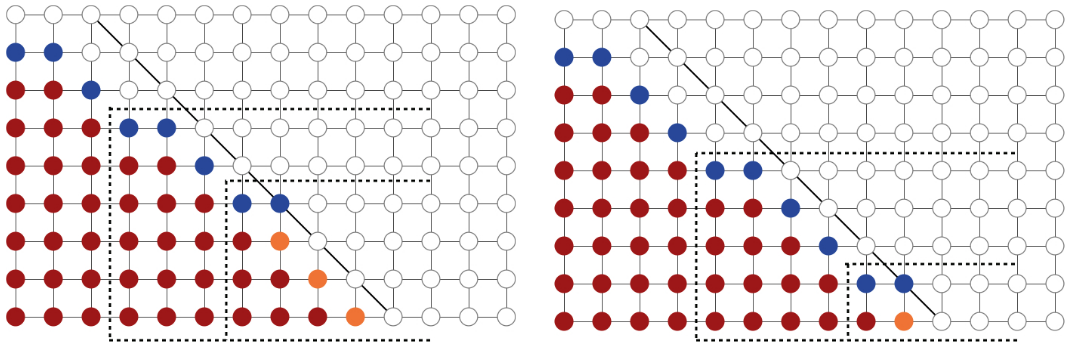
$$
\begin{aligned}
n(k+1) &= n(k) \\
&= \frac{k+2\omega}{\omega+1} \\
&= \frac{k+1+2\omega-1}{\omega+1} \\
&= \frac{k+1+2\omega-[(k-1) \mod (\omega+1)]}{\omega+1} \\
&= \frac{k+1+2\omega-[(k+1-2) \mod (\omega+1)]}{\omega+1}.
\end{aligned}
$$

Now, let time $k = i \mod (\omega+1)$ for $i \in \mathbb{Z}$ and $i \in [0, \omega-1] - \{1,2\}$. Suppose $n(k) = \frac{k+2\omega-[(k-2) \mod (\omega+1)]}{\omega+1}$, then by Remark 3.3.2 (3), we get

$$
\begin{aligned}
n(k+1) &= n(k) \\
&= \frac{k+2\omega-[(k-2) \mod (\omega+1)]}{\omega+1} \\
&= \frac{k+1+2\omega-[(k-2) \mod (\omega+1)]-1}{\omega+1} \\
&= \frac{k+1+2\omega-[(k+1-2) \mod (\omega+1)]}{\omega+1}
\end{aligned}
$$

Therefore, we proved that the formula satisfies all the conditions for $n(t)$. $\square$

Now we consider our containment approach. We use the same analogy as in the Section 3.2 for the quarter plane with resistance 1. We consider time k such that at $k+1$ all vertices become burned and ready to spread at time $k+2$.



(a) Network with resistance 2 (b) Network with resistance 3

Figure 3.3.3: Outline of containment strategy for quarter plane graphs.

Let a fire spread unprotected for k time step. We deploy one firefighter, so we use the same function

$$f(t) = \begin{cases} 1 & \text{for } t > k \\ 0 & \text{for } t \leq k. \end{cases}$$

Without loss of generality, suppose $k = 0 \mod (r+1)$. Then our containment strategy can be formulated as the following

$$\phi(k+i) = \begin{cases} (i-1, n(k) - \frac{i-1}{\omega+1}\omega) & \text{if } i = 1 \mod (\omega+1) \\ (i-1, n(k) - \frac{i-i \mod (\omega+1)}{\omega+1}\omega) & \text{if } i = m \mod (\omega+1) \text{ for } m \in \mathbb{Z} \cap [0, \omega] - \{1\} \end{cases}$$

Applying ϕ we get the following formula for the potential:

$$P(k+i) = \begin{cases} n(k) - \frac{i-1}{\omega+1}\omega & \text{if } i = 1 \mod (\omega+1) \\ n(k) - \frac{i-i \mod (\omega+1)}{\omega+1}\omega & \text{if } i = m \mod (\omega+1) \text{ for } m \in \mathbb{Z} \cap [0, \omega] - \{1\} \end{cases}$$

We shall consider for $k + i = 1 \mod (\omega + 1)$. Therefore, we can solve for i

$$\begin{aligned} P(k+i) &= n(k) - \frac{i-1}{\omega+1}\omega \\ n(k) &= \frac{i-1}{\omega+1}\omega \\ \frac{k + 2\omega - [(k-2) \mod (\omega+1)]}{\omega+1} &= \frac{i-1}{\omega+1}\omega \\ i &= \frac{k}{\omega} + 2 - 1 + \frac{1}{\omega} + 1 \\ i &= \frac{k+1}{\omega} + 2 \end{aligned}$$

Therefore, we conclude that if fire spreads unconstrainedly on a quarter plane with resistance ω, then applying our containment strategy, it takes $\frac{k+1}{\omega} + 2$ time steps to contain it.

4

Full Plane

4.1 Introduction

This section investigates a network, G, on a full plane. Fogarty [3] proved that one firefighter is not enough to contain a fire spread on a full plane. We show that fire spread on a grid with resistance level 1 can be contained with one firefighter.

4.2 Preliminaries

In order to understand investigation on a full plane, we can use many analogies from the Chapter 3 which investigated quarter plane. However, to understand further complexities entailed with the problem on a full plane, we shall introduce additional notation. Figure 4.1.1 shows a graph with unconstrained fire spread on a full network.

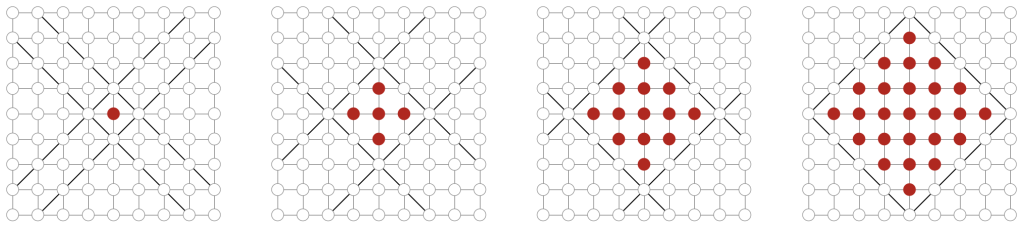

(a) time 1 (b) time 2 (c) time 3 (d) time 4

Figure 4.2.1: Unconstrained fire spread on a full plane with no resistance.

Length of potential fronts

Now, we introduce a new value used in [1], called the *length of a potential front*. Figure 4.2.2 shows an example of a scenario of fire spread on a full plane. We denote lengths of potential fronts as l_i for coresponding L_i. A unit length is equivalent to $\sqrt{2}$ in the Cartesian plane for the distance between two nodes diagonal from each other. Referring to the graph, we denote points A, B, C, D as points, where potential fronts intersect with each other. Note $A, B, C, D \in \mathbb{R} \times \mathbb{R}$. Then, we get that $l_1 = |AD|$, $l_2 = |AB|$, $l_3 = |BC|$ and $l_4 = |CD|$.

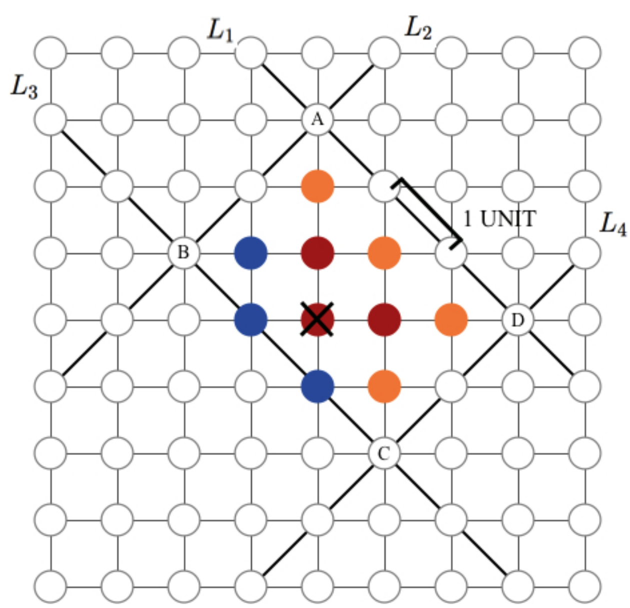


Figure 4.2.2: Example of fire spread scenario on graph G at time t.

Claim 4.2.1. [1] *Length of a potential front, denoted l_i, is given by the formula*

$$l_1 = l_3 = \frac{1}{2}(n_2 + n_4),$$
$$l_2 = l_4 = \frac{1}{2}(n_1 + n_3).$$

Proof. We provide a proof of the claim that was not included in [1]. We only show for l_1, as proofs for other lengths follow the exact same logic.

Consider the graph presented in Figure 4.2.2. Let $A = (x_a, y_a)$ and $D = (x_d, y_d)$. We know that

$$n_2 = -x_a + y_a \text{ and } n_4 = x_d - y_d.$$

We get that the length equals the following

$$l_1 = \frac{(x_d - x_a) + (y_a - y_d)}{2} = \frac{-x_a + y_a + x_d - y_d}{2} = \frac{n_2 + n_4}{2}.$$

□

The potential length also equals the number of nodes on the particular potential front. Note that half-values of bordering nodes are also considered.

Claim 4.2.2. *For any t, we have $P_i(t) \leq l_i(t)$.*

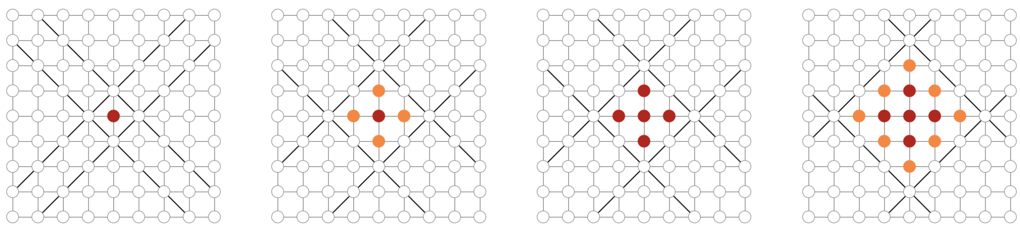
Proof. In an unconstrained spread, we have that $P_i(t) = l_i(t)$ for any t, which is the upper bound for the value of the potential. Otherwise, we would have firefighters placed on the potential front, and hence $P_i(t)$ would take a value less than $l_i(t)$. □

Claim 4.2.3. *For any t, we have $P_i(t) \geq n_i(t) - F_i(t)$.*

Proof. At time t, the lowest possible potential would be if $P_i(t) = n_i(t) - F_i(t)$, which means that all the firefighters that have been available since the beginning of the process are positioned on the fire front to minimize the number of unprotected nodes. □

4.3 Resistance Level 1

In this section, we look at a network with resistance level 1. Let G be a full plane with resistance 1. Figure 5.2.1 displays unconstrained fire spread on G. We prove that one firefighter is enough to contain the spread and we show our algorithm that automates fire containment in NetLogo program.



(a) time 1 (b) time 2 (c) time 3 (d) time 4

Figure 4.3.1: Unconstrained fire spread on a full plane with resistance 1.

Theorem 4.3.1. *Let G be a full plane with resistance 1. Then one firefighter is enough to contain the fire spread.*

Proof. We use NetLogo program to develop an algorithm for containment. We deploy one firefighter at each time step starting from time 1, so we have

$$f(t) = 1 \quad \text{for all } t \in \mathbb{N}.$$

At time 1, the graph is presented in Figure 4.3.2. We deploy our first firefighter at $(-1, 0)$.

Our containment strategy focuses on the following goals:

- the first firefighter is deployed at an intersection of potential fronts L_2 and L_3,
- we aim to contain the fire by protecting nodes along two adjacent potential fronts in order to "embrace" the fire spread,

- we attempt to keep one of the two adjacent potential fronts constant; in our case, the L_2,

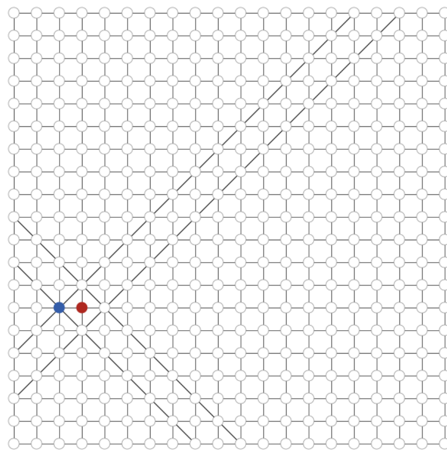


Figure 4.3.2: Graph G at time 1.

We assign "lead 1" and "lead 2" values to our firefighters. "Lead 1" is the last firefighter placed on L_1, and "Lead 2" is the last firefighter placed on L_2. Note that at time 1, the one deployed firefighter has both attributes, but after time 1, the attributes are passed on to the next deployed firefighters.

The algorithm for placing firefighters is guided by the following objectives:

1. "Lead 1" attempts to be at the top position from the farthest burning node in the upper direction (achieved at time 3 presented in Figure 4.3.3);

2. Once "lead 1" is assigned at the top, it attempts to prevent nodes on the potential front $L_2(t)$ at the time from catching on fire; i.e. the "lead 1" will be called to point at the next firefighter position if at the next time step the fire would spread on a node on L_2

3. If the fire is not yet endangering nodes on the potential front L_2, the algorithm calls "lead 2" to deploy firefighters

Our resulting algorithm essentially keeps one fire bound, i.e. the potential front L_2, in one static position and attempts to surround the fire with the "lead 2". If the conditions allow firefighters to contain the spread, the "lead 2" will eventually surround the fire from other three directions.

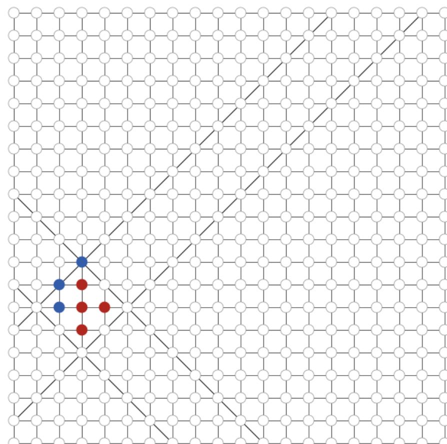
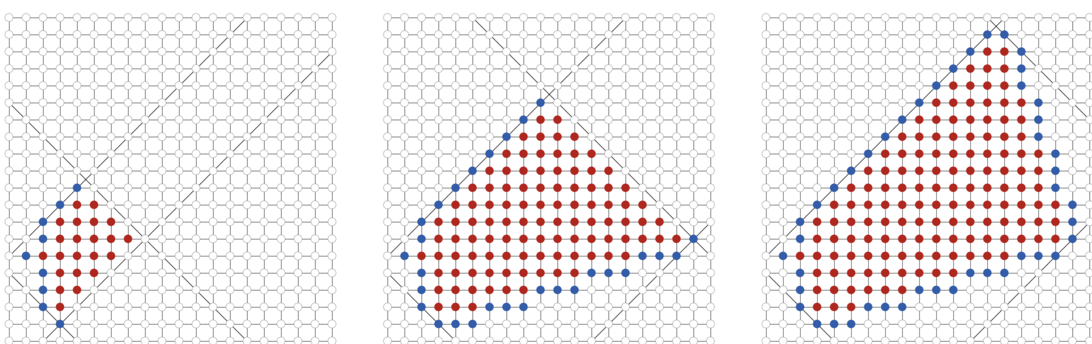


Figure 4.3.3: Graph G at time 3.

Applying the containment algorithm, we present several snapshots of the graph G at different times. At those time, we can also observe that the "lead 2" changes its direction of containment.



(a) time 9 (b) time 29 (c) time 45

Figure 4.3.4: Graph G with applied our containment algorithm.

The program showed that the fire can be contained with one firefighter. □

4.4 Discussion of Impossible Containment Proofs

In a network when all nodes are equally resistant, one firefighter is enough to contain a spread. We want to lay further ground for investigating bounds for the number of firefighters that are not enough to contain a spread. Some papers considered periodical functions for placing firefighters, i.e. an example is presented in the equation 4.4.1

$$f(t) = \begin{cases} 1 & \text{for } t \text{ odd} \\ 2 & \text{for } t \text{ even.} \end{cases} \tag{4.4.1}$$

This means that first one firefighter is deployed, then two, and then the pattern repeats. Feldheim [1] proved in his paper that this periodical equation 4.4.1 is not enough to contain a fire spread on a full plane with no resistance. We look at their lemmas to understand an approach to proving impossible containment. The notation is adjusted to match the terminology of this paper.

Lemma 4.4.1. [1] *If for all $t \in \mathbb{N}$, $l(t) \geq 2F(t) - 1$, then $P(t) > \frac{l(t)}{2}$.*

In other words, this lemma considers situations that our firefighters cannot "embrace" the fire from two directions, i.e. cover two adjacent fire fronts, then the fire will always spread indefinitely.

We consider this lemma to be crucial in proving that even given periodical functions for firefighters, the spread might never be contained. However, to prove it on networks with resistance level, relation between lengths of potential fronts and placing of firefighters needs to be further investigated.

We conjecture that on a full plane with resistance 1, periodical function placing one firefighter every two time steps is insufficient to contain a spread.

5
Concluding Discussion

5.1 NetLogo as the Simulation Program

In most prior researches the firefighter problem was modeled in JAVA. However, for this project's objectives, NetLogo environment proved to be very effective. The interface with visual presentation of different models of fire spread are very transparent and accurate. NetLogo is powerful tool in exploring effects of fire spread and possible containment strategies. Several drawbacks might be encountered when exploring graphs of structure that cannot be considered in grid framework.

5.2 Future Work

There are still many possible directions for future research of spread and containment modeling. Continuing the concepts of resistance investigated in this project, different resistance patterns could be explored. A more intuitive example could be considered in terms of spread of viruses and protecting people with vaccinations. If a model population had a strengthened immune system, how would it affect a contagion. Different parameters could be used, such as percentage of people having stronger immune system. For instance, if 50

% of the population was more immune to a virus, this could be modeled as every second node is assigned a higher resistance level. Geographical clusters of increased resistance could also be explored, which would probably entail incorporating statistical methods to form projections.

Appendix: NetLogo Code

```
breed [nodes node]

nodes-own [fire burned vulnerable resist firefighter t-burned potential lead]

undirected-link-breed [ fronts front ]

undirected-link-breed [ lines line ]

patches-own [resistance]

globals [level n1 n2 n3 n4 update start1 start2 fcount]

to setup-resistance-level

  set level 1     ;; resistance level 1

end


to setup

  clear-all

  reset-ticks

  setup-resistance-level

  setup-grid
```

```
  set-resistance
end

to manual
  click-put-firefighters
  update-states
end

to simulate-manual
  spread-fire
  ;set-potential-fronts
  tick
;  update-states
end

to simulate-contain
  spread-fire
  set-potential-fronts
  contain-fire
  update-states
  tick
  ;  if ticks = 50 [stop]

  let endcheck false
  if ticks > 3 [
    ask nodes with [lead = 1] [
```

```
      if distance one-of nodes with [lead = 2] < 1.5 [
        set endcheck true
      ]
    ]
  ]
  if endcheck [stop]
end

to contain-fire
  ifelse ticks = 0 [
    ask nodes-on patch -1 0 [
      put-firefighter
      set lead 1
      set start1 6
      set start2 0
    ]
  ][
  set update false
  ask nodes with [lead = 1] [
    ask nodes-at 1 0 [
      if fire = true and resist = 0 [
        ask nodes with [lead = 1] [
          screen-fire 1 start1 ((3 - start1) mod 8 + 1)
        ]
      ]    ]
    if update = true [stop]
```

```
    ask nodes-at 1 -1 [
      if not firefighter and not fire [
        ask nodes with [lead = 1] [
          screen-fire 1 start1 8
        ]
      ]
    ]
  ]
  if update = true [stop]
  ask nodes with [lead = 2] [
    screen-fire 2 start2 8
  ]
  ]
end

to screen-fire [leadval start maxrep] ; TURTLE procedure: the lead1 screens for fire sta
  let counter 0
  let indx ifelse-value (leadval = 1) [1][-1] ; give direction (lead1 - clockwise, lead2
  while [counter < maxrep] [
    check-fire leadval (start mod 8)
    if update [stop]
    set start (start + indx)
    set counter (counter + 1)
    ]
  if update = false [
    user-message "error in screen-fire"]
```

```
end

to check-fire [leadval num] ; TURTLE procedure: given the lead and a number, checks the
  let value ifelse-value (leadval = 1) [true] [false]
  if num = 0 [
    ask nodes-at -1 1 [
      if fire = true [
        ifelse value [
          place-f-at 1 0 -1
          set start1 5
        ][
        place-f-at 2 1 0
        set start2 3 ]]
    ]
  ]
  if num = 1 [
    ask nodes-at 0 1 [
      if fire = true [
        ifelse value [
          place-f-at 1 -1 0
          set start1 6
        ][
        place-f-at 2 1 0
        set start2 4
        ]]]
  ]
```

```
if num = 2 [
  ask nodes-at 1 1 [
    if fire = true  [
      ifelse value [
        place-f-at 1 -1 0
        set start1 7
      ][
      place-f-at 2 0 -1
      set start2 5
      ]]]
]
if num = 3 [
  ask nodes-at 1 0 [
    if fire = true  [
      ifelse value [
        place-f-at 1 0 1
        set start1 0
      ][
      place-f-at 2 0 -1
      set start2 6
      ]]]
]
if num = 4 [
  ask nodes-at 1 -1 [
    if fire = true [
      ifelse value [
```

```
        place-f-at 1 0 1
        set start1 1
      ][
      place-f-at 2 -1 0
      set start2 7
      ]]]
]
if num = 5 [
  ask nodes-at 0 -1 [
    if fire = true  [
      ifelse value [
        place-f-at 1 1 0
        set start1 2
      ][
      place-f-at 2 -1 0
      set start2 0
      ]]]
]
if num = 6 [
  ask nodes-at -1 -1 [
    if fire = true  [
      ifelse value [
        place-f-at 1 1 0
        set start1 3
      ][
      place-f-at 2 0 1
```

```
        set start2 1
        ]]]
  ]
  if num = 7 [
    ask nodes-at -1 0 [
      if fire = true  [
        ifelse value [
          place-f-at 1 0 -1
          set start1 4
        ][
        place-f-at 2 0 1
        set start2 2
        ]]]
  ]
  if update [
    ifelse any? nodes with [lead = 2] [
    set lead 0
    ][
    set lead 2
    ]
  ]
end

to place-f-at [leadval  ddx ddy]  ; TURTLE procedure : places a firefighter at the relat
  ask nodes-at ddx ddy [
    put-firefighter
```

```
    set update true
    set lead leadval
  ]
end


to put-firefighter ; TURTLE procedure to position a firefighter at the node
  if fire [
    show "ERROR: node with fire and firefighter"
    user-message "ERROR: cannot put firefighter (node already on fire)"]
  set firefighter true
  set shape "circle"
  set color blue
end


;; position firefighters with mouse clicks ;;
to click-put-firefighters
  if mouse-down? [
    ask patch (round mouse-xcor) (round mouse-ycor) [
      ask nodes-here [
        put-firefighter
      ]
    ]
  ]
end
```

```
;; updates to state of nodes - graph presentation ;;

to spread-fire
  ask fronts [die]
  ask turtles [set potential false] ; everytime reset the potential
  ifelse ticks = 0 [ ; fire initially breaks out at the origin
    ask patch 0 0 [
      ask nodes-here [
        set fire true
        set resist 0
      ]
    ]
  ]
  [ ask nodes with [ fire = true and resist = 0 ] [
    ask line-neighbors with [not firefighter] [
      ifelse vulnerable = true [
        set fire true
        set t-burned ticks
      ]
      ;; mark the time it was burned so avoid doubled fire spred ;;
      [ if t-burned < ticks [
        set resist resist - 1]
      ]
    ]
  ]
```

```
  ]
end

to update-states
  ask nodes [
    ;; eliminate negative values for resistance level ;;
    if resist < 0 [
      set resist 0
    ]
    ;; nodes no longer vulnerable ;;
    if fire = true or firefighter = true [
      set vulnerable false
    ]
    ;; 'burned' state of nodes ;;
    if fire = true and resist = 0 [
      set burned true
      set color 14
      set shape "circle"
    ]
    ;; 'burning' state of nodes ;;
    if fire = true and resist > 0 [
      set color 26
      set shape "circle"
    ]
    if firefighter = true and fire = true [
      user-message "ERROR: node both burning and protected!"
```

```
      show self
      show "The time it happened"
      show ticks
    ]
  ]
end

to set-resistance
  ask patches [
    set resistance true
    ask nodes-here [
      set resist level
      set color 5
    ]
  ]
end

to setup-grid
  ask patches [
    sprout-nodes 1 [
      setxy pxcor pycor
      set shape "node"  ; shape of nodes
      set size .50
      set color 0 ;54
                ; initial state of all nodes
      set fire false
```

```
      set burned false
      set firefighter false
      set resist 0
      set vulnerable true
      set lead 0
      set t-burned 0 ; time at which a node becomes infected
      set potential false
    ]
    set pcolor 9.9 ; set color of patches
    set resistance false
  ]
  ask turtles [
    let neighbor-nodes turtle-set [turtles-here] of neighbors4
    create-lines-with neighbor-nodes [
      set color 3
    ]
  ]
end

to set-potential-fronts
    ask nodes with [fire] with-max [xcor + ycor] [set n1 ( xcor + ycor + 1 )]
    ask nodes with [fire] with-max [ycor - xcor] [set n2 ( ycor - xcor + 1 )]
    ask nodes with [fire] with-max [(- xcor) - ycor] [set n3 ( (- xcor) - ycor + 1 )]
    ask nodes with [fire] with-max [xcor - ycor] [set n4 ( xcor - ycor + 1)]
  ask nodes with [ xcor + ycor = n1 ] [
    set potential true
```

```
  if any? nodes-at -1 1 [
    create-fronts-with nodes-at -1 1]
  if any? nodes-at 1 -1 [
    create-fronts-with nodes-at 1 -1]
]
ask nodes with [ ycor - xcor = n2 ] [
  set potential true
  if any? nodes-at -1 -1 [
    create-fronts-with nodes-at -1 -1]
  if any? nodes-at 1 1 [
    create-fronts-with nodes-at 1 1]
]
ask nodes with [ (- xcor) - ycor = n3 ] [
  set potential true
  if any? nodes-at -1 1 [
    create-fronts-with nodes-at -1 1]
  if any? nodes-at 1 -1 [
    create-fronts-with nodes-at 1 -1]
]
ask nodes with [ xcor - ycor = n4 ] [
  set potential true
  if any? nodes-at 1 1 [
    create-fronts-with nodes-at 1 1]
  if any? nodes-at -1 -1 [
    create-fronts-with nodes-at -1 -1]
]
```

```
  ask fronts [
    set thickness 0.05
    set color black
  ]
end
```

Bibliography

[1] Ohad N Feldheim and Rani Hod, *3/2 firefighters are not enough*, CoRR (2010).

[2] A.S. Finbow, B.L. Hartnell, Q. Li, and K. Schmeisser, *On Minimizing the Effects of Fire on a Virus on a Network*, J. Combin. Math. Combin. Comput **33** (2000), 311-322.

[3] P. Fogarty, *Catching The Fire On Grids*, M.Sc. Thesis, Department of Mathematics, University of Vermont, USA, 2003.

[4] Stephen G. Hartke, *Graph-Theoretic Models of Spread and Competition*, Ph.D. dissertation, Rutgers University, 2004.

[5] B.L. Hartnell, *Firefighter! an application of domination*, 24th Manitoba Conference on Combinatorial Mathematics and Computing, University of Manitoba in Winnipeg, Canada, 1995.

[6] P. Wang and S. Moeller, *Fire Control On Graphs*, J. Comb. Math. & Comb. Computing **41** (2002), 19–34.

[7] Robert J. Wilson, *Introduction to Graph Theory*, Pearson, October 2012.