Chapter 3

Response Strategies in Deterministic Models of Spread: Vaccination and Firefighting

3.1 Introduction

Traditionally, epidemiological models assume that the population being studied is well-mixed in the sense that any pair of individuals are just as likely to come in contact and transmit a disease as any other. This "mean-field" approximation, as physicists call it, appears in such models as the SIR model (which tracks the numbers of Susceptibles, Infected, and Recovered), and this simplifying assumption permits exact solutions using ordinary differential equations. For some diseases and settings where the well-mixing assumption is reasonable, such as influenza in an elementary school, these models come quite close to observed data.

However, for other diseases and contexts, the spatial component is much more important. The spread of rabies in rabbit populations in Switzerland is one such example. In situations where the spatial component has a strong geometric structure, usually resulting from geographic locations, partial differential equation models have been successful in modeling the "wave front" of the disease spread. When the spatial component does not have a strong geometric structure, as is the case with AIDS and other sexually-spread diseases, the relationships between people must be considered on a pair-by-pair basis [12, 1]. The mathematical structures of graphs are ideally suited to encode these relationships, where vertices in a graph represent individuals, and edges represent the potential for transmission of the disease between two individuals.

One avenue of exploring graph-based models has been the use of agent-based computer simulations. In these simulations, the relationships and health of each individual (or "agent") is determined at each point in time [2]. Los Alamos' EpiSIMS project [3] is an example of a very large model, simulating the interactions of 1.6 million people in the greater Portland, Oregon, area. Agent-based simulations are very useful for experimenting with models and suggesting what the behavior of the model is. However, it is difficult to use the simulations to prove precise statements about the behavior. The approach followed here is to look at these problems from a more combinatorial and graph-theoretic perspective.

In this chapter we focus on a deterministic process and how it behaves when various interventions are occurring. This is particularly relevant in disease spread processes, where vaccinations and quarantines are being used to contain a disease outbreak. From a graph theoretic perspective, such interventions have the effect of a vertex or edge cut. However, the dynamic nature of the disease spread makes the problem more difficult.

Epidemiologists have proposed several spread mechanisms based on the biological properties of different diseases. These mechanisms determine the rate and likelihood of transferring the disease from an infected individual to a susceptible individual. In this chapter we consider the most simple spread mechanism: that of a perfectly contagious disease with no cure, where vertices adjacent to infected vertices become infected at every discrete time step and, once infected, remain infected from then on. The response allowed is only a limited number of vaccinations of non-infected vertices. Specifically, let G be a connected graph where the vertices represent people and the edge uv indicates that persons u and v would transmit a disease from one person to another if one person became infected. At time $t = 0$, some outbreak of disease occurs at several root vertices. Public health officials immediately respond, vaccinating the vertices $a_{1,1}, a_{1,2}, \ldots, a_{1,c_1}$ at time $t = 1$. The disease then spreads to every non-vaccinated neighbor of an infected vertex. There is another set of vaccinations $a_{2,1}, \ldots, a_{2,c_2}$ at time $t = 2$, and the disease spreads again. This process continues until the disease can no longer spread; in other words, that all of the neighbors of infected vertices are either themselves infected or vaccinated. The main question we will investigate is finding an optimal strategy for vaccinating in order to minimize the total number of infected vertices. We will be primarily interested in the situation when there is only one initially infected root vertex, and there are exactly f vaccinations allowed per time step. In section 3.2 we examine the case when G is a grid, and in section 3.4.1 we discuss an approximation technique for the problem on trees. In section 3.3 we present a proof that this problem is NP-complete for general graphs, and in the last section we discuss future work.

The model of disease spread just presented is equivalent to a model of fire spread introduced by Hartnell [9]. In this model, an outbreak of fire starts at the root vertices at time $t = 0$. In response, firefighters are placed at the vertices $a_{1,1}, a_{1,2}, \ldots, a_{1,c_1}$ at time $t = 1$, where the firefighters defend or protect each vertex from the spreading fire. The fire then spreads from burning vertices to non-defended neighbors. Firefighters are again deployed to defend the vertices $a_{2,1}, \ldots, a_{2,c_2}$ at time $t = 2$ (the vertices $a_{1,1}, a_{1,2}, \ldots, a_{1,c_1}$ remain defended), and the fire spreads again. The process continues until the fire can no longer spread. We say that the fire outbreak is *contained* after t time steps if there is some finite time t such that after the disease spreads during time t, only a finite number of vertices are burnt and the disease can no longer spread. The motivating question is again to find an optimal sequence of defended vertices that minimizes the total number of burnt vertices.

When presenting our results, we will use the terminology of firefighters. During the t^{th} time step for $t > 0$, firefighters are deployed and then the fire spreads. If we describe the state of vertices at the *beginning of the t^{th} time step*, we mean *before* the firefighters are deployed during the t^{th} time step. If we describe the state of vertices at the *end of the t^{th} time step*, or equivalently, at the *end of t time steps,* we mean *after* the fire has spread during the t^{th} time step. A firefighter may defend neither a burnt vertex nor a previously defended vertex. Once fire has spread to a vertex v, we say that v is a *burnt* vertex. After being burnt or defended, a vertex remains in that state until the process ends. In addition to the burnt and defended vertices, we say that a vertex v is *saved* at the end of the t^{th} time step if there is no path from v to the root consisting only of burnt and non-defended vertices at the end of the t^{th} time step. Thus, our motivating question is equivalent to maximizing the number of saved vertices.

Several results are known about this model for various classes of graphs. Wang and Moeller [13] studied grids and other product graphs. They determined that two firefighters per time step is sufficient to contain a fire outbreak in a two dimensional square grid, and conjectured that $2d-1$ firefighters are necessary to contain a fire outbreak in a d dimensional square grid. We prove this conjecture in section 3.2. Fogarty [6] showed that two firefighters suffice in the two dimensional square lattice to contain any finite outbreak of fire where an arbitrarily large but finite number of vertices are initially on fire. However, we prove that for any fixed number f of firefighters, there is a finite outbreak of fire in which f firefighters per time step are insufficient to contain the outbreak.

MacGillivray and Wang [11] showed that the problem of determining an optimal sequence of firefighter placements that saves the most vertices is NP-complete for general graphs. We present a different proof of NP-completeness in section 3.3 that uses graphs of smaller average degree, a more realistic assumption for the disease application. Finbow, King, MacGillivray, and Rizzi [5] show that the firefighter problem is NP-complete for trees of maximum degree three. MacGillivray and Wang also presented bounds and algorithms for trees and square grids. Hartnell and Li [10] showed that the greedy algorithm on trees always saves at least 1/2 as many vertices as an optimal sequence of firefighter placements. Finbow, Hartnell, Li, and Schmeisser [4] determine the graphs that have the lowest number of expected burnt vertices when the initial root vertex where the fire outbreak begins is random.

3.2 Grids[1]

Grids are a natural class of graphs to consider both disease and fire spread on since they are often used to represent geographic areas. We consider here the infinite d-dimensional square grids $\mathbb{L}^d$. The vertices of $\mathbb{L}^d$ are the points of $\mathbb{R}^d$ with integer coordinates, and x is adjacent to y if and only if x is distance 1 from y in the usual Euclidean ℓ_2 metric.

3.2.1 Three and Higher Dimensional Square Grids

Wang and Moeller proved in [13] that an outbreak starting at a single point in a regular graph of degree r can be contained with if $r-1$ firefighters can be deployed per time step.

[1]This section contains joint work with Mike Develin.

Specifically, for the d dimensional square grid $\mathbb{L}^d$, $2d-1$ firefighters suffice to contain an outbreak starting at a single point. They conjectured that this bound is tight, and we present a proof of this conjecture here.

Wang and Moeller observed that at least two firefighters per time step are needed for containment in $\mathbb{L}^2$, and Fogarty showed in [6] that at least three firefighters per time step are needed to contain the outbreak. Her main theorem involves a "Hall-type condition" which we strengthen here in Theorem 3.2. First we state some definitions.

Definition 3.1. Let D_k denote the set of vertices in a rooted graph G that are distance k from the root vertex r. Let r_k denote the number of firefighters in $D_{k+1}, D_{k+2}, \ldots$ at the end of the k^{th} time step. These firefighters can be thought of as "reserve" firefighters since they are not adjacent to the fire when deployed. We define r_0 to be 0. Let $B_k \subseteq D_k$ denote the number of burned vertices in D_k at the end of the k^{th} time step.

Theorem 3.2. *Let G be a rooted graph, h a positive integer, and $a_0, a_1, \ldots, a_h$ positive integers each at least f such that the following holds:*

1. *Every $A \subseteq D_0$, $A \neq \emptyset$, satisfies $|N(A) \cap D_1| \geq |A| + a_0$.*
2. *For $1 \leq k \leq h$, every $A \subseteq D_k$ where $|A| \geq 1 + \sum_{i=0}^{k-1}(a_i - f)$ satisfies $|N(A) \cap D_{k+1}| \geq |A| + a_k$.*
3. *For $k > h$, every $A \subseteq D_k$ such that $|A| \geq 1 + \sum_{i=0}^{h}(a_i - f)$ satisfies $|N(A) \cap D_{k+1}| \geq |A| + f$.*

Suppose that at most f firefighters per time step are deployed. Then

$$|B_n| \geq \begin{cases} 1 & \text{if } n = 0, \\ 1 + r_n + \sum_{i=0}^{n-1}(a_i - f) & \text{if } 1 \leq n \leq h+1, \\ 1 + r_n + \sum_{i=0}^{h}(a_i - f) & \text{if } n > h+1, \end{cases} \tag{3.1}$$

regardless of the sequence of firefighter placements. Specifically, f firefighters per time step are insufficient to contain an outbreak that starts at the root vertex.

Proof. Let p_{n+1} denote the number of firefighters placed in D_{n+1} at time $n+1$, and let $p_{\leq n}$ denote the number of reserve firefighters placed in D_{n+1} during time steps $1, \ldots, n$. Note that

$$r_{n+1} \leq (r_n - p_{\leq n}) + (f - p_{n+1}) = r_n + f - p_{n+1} - p_{\leq n}. \tag{3.2}$$

This follows since $r_n - p_{\leq n}$ is the number of firefighters placed in $D_{n+2}, D_{n+3}, \ldots$ for times $1, \ldots, n$, and at most $f - p_{n+1}$ firefighters are available to be placed in $D_{n+2}, D_{n+3}, \ldots$ at time $n+1$. Strict inequality occurs if a firefighter is placed in D_k for $k < n+1$ at time $n+1$.

We prove (3.1) by induction on n. For $n = 0$, $|B_0| = 1$ holds trivially. We assume the result holds for n, $0 \leq n \leq h$, and prove the result for $n+1$. By inductive hypothesis,

$$|B_n| \geq \begin{cases} 1 & \text{if } n = 0, \\ 1 + r_n + \sum_{i=0}^{n-1}(a_i - f) & \text{if } 1 \leq n \leq h, \end{cases} \tag{3.3}$$

and so by hypotheses 1 and 2,

$$|N(B_n) \cap D_{n+1}| \geq |B_n| + a_n. \tag{3.4}$$

Thus,

$$\begin{aligned} |B_{n+1}| &= |N(B_n) \cap D_{n+1}| - p_{n+1} - p_{\leq n} \\ &\geq |B_n| + a_n - p_{n+1} - p_{\leq n}, \text{ by (3.4)}, \\ &\geq 1 + r_n + \sum_{i=0}^{n-1}(a_i - f) + a_n - p_{n+1} - p_{\leq n}, \text{ by (3.3)}, \\ &= 1 + (r_n + f - p_{n+1} - p_{\leq n}) + \sum_{i=0}^{n-1}(a_i - f) + (a_n - f) \\ &\geq 1 + r_{n+1} + \sum_{i=0}^{n}(a_i - f), \text{ by (3.2)}. \end{aligned}$$

This proves (3.1) for $0 \leq n \leq h+1$.

We now prove (3.1) for $n \geq h+1$ using induction on n. Note that (3.1) holds for $n = h+1$ from above. We thus assume (3.1) holds for $n \geq h+1$, and we prove the result for $n+1$. By inductive hypothesis,

$$|B_n| \geq 1 + r_n + \sum_{i=0}^{h}(a_i - f), \tag{3.5}$$

and so by hypothesis 3, (3.4) holds for $n > h$. Thus,

$$\begin{aligned}
|B_{n+1}| &= |N(B_n) \cap D_{n+1}| - p_{n+1} - p_{\leq n} \\
&\geq |B_n| + f - p_{n+1} - p_{\leq n}, \text{ by (3.4)}, \\
&\geq 1 + r_n + \sum_{i=0}^{h} (a_i - f) + f - p_{n+1} - p_{\leq n}, \text{ by (3.5)}, \\
&= 1 + (r_n + f - p_{n+1} - p_{\leq n}) + \sum_{i=0}^{h} (a_i - f) \\
&\geq 1 + r_{n+1} + \sum_{i=0}^{h} (a_i - f), \text{ by (3.2)}.
\end{aligned}$$

□

We now turn our attention to square lattices of dimension three and higher.

Definition 3.3. The orthants of $\mathbb{R}^d$ are the 2^d regions defined by the hyperplanes $x_i = -1/2$ in $\mathbb{R}^d$, $i = 1, \ldots, d$. Let the orthants in $\mathbb{L}^d$ be the subsets of vertices that lie in each orthant of $\mathbb{R}^d$. Thus, the j^{th} coordinates of all the vectors in a given orthant of $\mathbb{R}^d$ are all non-negative or are all negative, for $j = 1, \ldots, d$. Let D_k^+ denote the vertices of $D_k \subseteq \mathbb{L}^d$ in the orthant whose elements are all non-negative.

Let $v = (v_1, v_2, \ldots, v_d)$ be an element of $D_k \subseteq \mathbb{L}^d$. Let $c_i(v)$ denote v_i, and for a set $A \subseteq D_k$ define $A_r^i = \{v \in A : c_i(v) = r\}$. Let $v_{\to i}$ denote $(v_1, v_2, \ldots, v_i', v_{i+1}, \ldots, v_d) \in D_{k+1}$, where $v_i' = v_i + 1$ if $v_i \geq 0$ or $v_i' = v_i - 1$ if $v_i < 0$. Thus, $v_{\to i}$ is in the same orthant as v.

Lemma 3.4. *In $\mathbb{L}^d$ for $d \geq 3$, if $A \subseteq D_k$ where $|A| \geq 2d - 2$, then $|N(A) \cap D_{k+1}| \geq |A| + 2d - 2$.*

Proof. Given any nonempty set $A \subseteq D_k \subseteq \mathbb{L}^d$ completely contained in one orthant, we will show that

$$|N(A) \cap D_{k+1}| \geq |A| + d - 1, \text{ for any } d. \tag{3.6}$$

We form a set $B \subseteq N(A) \cap D_{k+1}$ in the following way:

1. For each $v \in A$, add $v_{\to 1}$ to B.

2. For each $2 \leq j \leq d$, let r_j be the value of the j^{th} coordinate of elements of A that is greatest in absolute value. For each $v \in A_{r_j}^j$, add $v_{\to j}$ to B.

Each vector added to B in step 1 is unique, and each vector added to B in step 2 is also unique since the j^{th} coordinate was chosen to be maximum. Thus, $|N(A) \cap D_{k+1}| \geq |B| \geq |A| + d - 1$.

Let $A \subseteq D_k \subseteq \mathbb{L}^d$. If A is not completely contained in one orthant, then let A be partitioned as

$$A = A_1 \cup A_2 \cup \cdots \cup A_q,$$

where each A_ℓ is in a different orthant $\mathcal{O}_\ell$. By (3.6), $|N(A_\ell) \cap D_{k+1}| \geq |A_\ell| + d - 1$. Note also that the corresponding sets B_ℓ in the proof above for A_ℓ do not overlap since they are in different orthants. Hence,

$$\begin{aligned} |N(A) \cap D_{k+1}| &\geq \sum_{\ell=1}^{q} |N(A_\ell) \cap \mathcal{O}_\ell \cap D_{k+1}| \\ &\geq \sum_{\ell=1}^{q} [|A_\ell| + d - 1] \\ &\geq |A| + 2d - 2. \end{aligned}$$

Thus, we may assume that A is completely contained in one orthant, and, without loss of generality, we assume that all coordinates of elements of A are non-negative.

We now proceed to prove the lemma by induction on d. Let $A \subseteq D_k^+ \subseteq \mathbb{L}^d$, where $|A| \geq 2d - 2$. Suppose that $d = 3$. Let n_i denote the number of nonempty A_r^i, or, equivalently, the number of distinct i^{th} coordinates of elements of A. Let i' be a coordinate where n_i is maximized. We claim that $n_{i'} \geq 3$. If $n_{i'}$ is 1, then A contains only one element, which is a contradiction since $|A| \geq 2d - 2 = 6$. If $n_{i'}$ is 2, then each coordinate has only two different values it can assume. However, the sum of the coordinates must remain k. It is straightforward to verify that the maximum number of elements in A is 3, which contradicts the fact that$|A| \geq 2d - 2 = 6$. Thus, $n_{i'} \geq 3$.

For each r where $A_r^{i'}$ is nonempty, form a set $\widehat{A_r^{i'}} \subseteq D_{k-r}^{d-1} \subseteq \mathbb{L}^{d-1}$ by eliminating the i' coordinate of each element in $A_r^{i'}$. By (3.6), $\left|N(\widehat{A_r^{i'}}) \cap D_{k-r+1}^{d-1}\right| \geq \left|\widehat{A_r^{i'}}\right| + d - 2$. For each v in $N(\widehat{A_r^{i'}}) \cap D_{k-r+1}^{d-1}$, form an element $\widetilde{v}$ in $N(A_r^{i'}) \cap D_{k+1}^d$ by inserting r as the i' coordinate. Notice that these elements are distinct when the i' coordinates are distinct. Let m be the maximum r such that $A_r^{i'}$ is nonempty, or equivalently, the largest i' coordinate. For each

$v \in A^{i'}_m$, we also have $v_{\to i'} \in N(A) \cap D_{k+1}$, and these vectors are distinct from any formed above because the i' coordinate is larger. Thus,

$$\begin{aligned} |N(A) \cap D_{k+1}| &\geq \sum_{r : A^{i'}_r \neq \emptyset} \left(\left|A^{i'}_r\right| + d - 2 \right) + \left|A^{i'}_m\right| \\ &\geq |A| + n_{i'}(d-2) + \left|A^{i'}_m\right|. \end{aligned} \tag{3.7}$$

Since $\left|A^{i'}_m\right| \geq 1$, (3.7) implies that

$$|N(A) \cap D_{k+1}| \geq |A| + 3d - 5, \tag{3.8}$$

and when $d = 3$,

$$|N(A) \cap D_{k+1}| \geq |A| + 4 = |A| + 2d - 2.$$

Now suppose that $d > 3$. Again let n_i denote the number of nonempty A^i_r, and let i' be a coordinate where n_i is maximized. If $n_{i'} \geq 3$, then using the same construction as in the $d = 3$ case, we have (3.8), and since $d > 3$, $|N(A) \cap D_{k+1}| \geq |A| + 2d - 2$. If $n_{i'} = 1$, then A contains only one element, which is a contradiction since $|A| \geq 2d - 2 \geq 4$. We are thus left with the case $n_{i'} = 2$. Let m be the maximum r such that $A^{i'}_r$ is nonempty, or equivalently, the largest i' coordinate of elements of A, and let $r' \neq m$ be the minimum value of r where $A^{i'}_r$ is nonempty. If $\left|A^{i'}_m\right| \geq 2$, then using the same construction as in the $n_{i'} \geq 3$ case, we have by (3.7)

$$\begin{aligned} |N(A) \cap D_{k+1}| &\geq |A| + n_{i'}(d-2) + \left|A^{i'}_m\right| \\ &\geq |A| + (2d-4) + 2, \text{ since } \left|A^{i'}_m\right| \geq 2, \\ &\geq |A| + 2d - 2. \end{aligned}$$

If $\left|A^{i'}_m\right| = 1$, then we again use the construction from the $n_{i'} \geq 3$ case. However, $\left|\widehat{A^{i'}_{r'}}\right| \geq 2d-3$, so by induction, $\left|N(\widehat{A^{i'}_{r'}}) \cap D^{d-1}_{k-r'+1}\right| \geq \left|\widehat{A^{i'}_{r'}}\right| + 2d - 4$. Here, the notation D^{d-1}_z means the set $D_z \subseteq \mathbb{L}^{d-1}$, emphasizing the dimension of $\mathbb{L}^{d-1}$. For each v in $N(\widehat{A^{i'}_{r'}}) \cap D^{d-1}_{k-r'+1}$, form an element $\widetilde{v}$ in $N(A^{i'}_{r'}) \cap D^d_{k+1}$ by inserting r' as the i' coordinate. Additionally, for the single vector $v \in A^{i'}_m$ and $1 \leq j \leq d$, $v_{\to j} \in N(A) \cap D_{k+1}$, and these vectors are distinct from those formed above because the i' coordinate is larger. Thus,

$$\begin{aligned}|N(A) \cap D_{k+1}| &\geq \left(\left|A_{r'}^{i'}\right| + 2d - 4\right) + d \\ &= |A| + 3d - 3, \text{ since } \left|A_{r'}^{i'}\right| = |A| + 1, \\ &\geq |A| + 2d - 2, \text{ since } d > 3. \end{aligned}$$ □

Lemma 3.5. *In* $\mathbb{L}^d$ *for* $d \geq 3$*, if* $A \subseteq D_1$ *where* $|A| \geq 2$*, then* $|N(A) \cap D_2| \geq |A| + 4d - 6$.

Proof. Let $A \subseteq D_1 \subseteq \mathbb{L}^d$ where $|A| \geq 2$. Every vector $v \in A$ is of the form $(0, \ldots, x_i, \ldots, 0)$, where $x_i = \pm 1$. Each vector v in A has $2(d-1)$ neighbors in D_2 formed by replacing each of the zero coordinates in v with ± 1, and one neighbor formed by replacing 1 in the i^{th} coordinate with 2 or replacing -1 with -2. If v and v' are vectors of A with nonzero entries in different coordinates, then v and v' share exactly one neighbor in D_2. If v and v' have nonzero entries in the same coordinate, then v and v' share no neighbors in D_2. Thus,

$$\begin{aligned}|N(A) \cap D_2| &\geq |A|\left(2(d-1)+1\right) - \binom{|A|}{2} \\ &= |A|\left(2d - \frac{|A|}{2} - \frac{1}{2}\right) \\ &\geq |A| + |A|\left(2d - \frac{|A|}{2} - \frac{3}{2}\right).\end{aligned}$$

It is straightforward to use calculus to verify that

$$|A|\left(2d - \frac{|A|}{2} - \frac{3}{2}\right) \geq 4d - 6,$$

where $d \geq 3$ and $2 \leq |A| \leq 2d$, and so

$$|N(A) \cap D_2| \geq |A| + 4d - 6.$$ □

Theorem 3.6. *In* $\mathbb{L}^d$*,* $2d-1$ *firefighters are needed to contain an outbreak of fire starting at a single vertex.*

Proof. Since $\mathbb{L}^d$ is vertex transitive, we may assume that the root vertex where the fire outbreak starts is the origin. We use Theorem 3.2 with $f = 2d-2$, $h = 1$, $a_0 = 2d-1$, and $a_1 = 4d-6$. The one element set D_0 has $2d$ neighbors in D_1 so hypothesis 1 of Theorem 3.2 holds, Lemma 3.5 shows hypothesis 2 of Theorem 3.2 holds for $k = 1$, and Lemma 3.4 shows hypothesis 3 holds for $k > 1$. By Theorem 3.2, $2d-2$ firefighters are insufficient to contain an outbreak starting at the origin. $\square$

Fogarty also showed in [6] that two firefighters suffice in $\mathbb{L}^2$ to contain any finite outbreak of fire where an arbitrarily large but finite number of vertices are initially on fire. However, we prove for $\mathbb{L}^d$ where $d \geq 3$ that for any fixed number f of firefighters, there is a finite outbreak of fire in which f firefighters per time step are insufficient to contain the outbreak.

First we establish the following lemma. Essentially, the lemma says that if we have a "front" of x elements, then it will grow outwards by at least $\Omega(\sqrt{x})$ in the next time step.

Lemma 3.7. *Let f be any positive integer. If $A \subseteq D_k^+ \subseteq \mathbb{L}^3$ where $|A| > \frac{1}{2}(f-1)(f-2)$, then $\left|N(A) \cap D_{k+1}^+\right| \geq |A| + f$.*

Proof. Let $A \subseteq D_k^+ \subseteq \mathbb{L}^3$ be a set where $|A| > \frac{1}{2}(f-1)(f-2)$. The elements of $B := \{v_{\rightarrow 1} : v \in A\}$ are distinct vertices in $N(A) \cap D_{k+1}^+$, and the set B has cardinality equal to $|A|$. Therefore, it suffices to show that if $|A| > \frac{1}{2}(f-1)(f-2)$, then there are at least f distinct elements of the form $v_{\rightarrow j}$ which are not elements of B, where $v \in A$ and $j \in \{2, 3\}$.

Let m be the largest first coordinate of elements of A, and let t be the smallest first coordinate of elements of A. Recall that the sets A_r^1, $r = t, t+1, \ldots, m$, partition A. Let σ_r equal $\left|A_r^1\right|$, so that $\sum_{r=t}^m \sigma_r = |A|$. Note that $\sigma_t, \sigma_m > 0$.

Suppose some σ_r is equal to zero, where $t < r < m$. Then A is partitioned into the sets A_1 consisting of all elements of A with first coordinate greater than r and A_2 consisting of all elements of A with first coordinate less than r. Clearly $N(A_1) \cap N(A_2) \cap D_{k+1}^+ = \varnothing$. Define $A_1' := \{v_{\rightarrow 2} : v \in A_1\}$ and $A_2' := \{v_{\rightarrow 1} : v \in A_2\}$, so that A_1' and A_2' are subsets of D_{k+1}^+. Since A_1' is simply a translate of A_1 by 1 in the first coordinate, $N(A_1') \cap D_{k+2}^+$ is a translate of $N(A_1) \cap D_{k+1}^+$ by 1 in the first coordinate. Similarly, $N(A_2') \cap D_{k+2}^+$ is a translate of $N(A_2) \cap D_{k+1}^+$ by 1 in the second coordinate. Thus, we have that

$$\begin{aligned}\left|N(A_1' \cup A_2') \cap D_{k+2}^+\right| &\leq \left|N(A_1') \cap D_{k+2}^+\right| + \left|N(A_2') \cap D_{k+2}^+\right| \\ &= \left|N(A_1) \cap D_{k+1}^+\right| + \left|N(A_2) \cap D_{k+1}^+\right| \\ &= \left|N(A) \cap D_{k+1}^+\right|,\end{aligned}$$

where the last equality follows since $N(A_1) \cap D^+_{k+1}$ and $N(A_2) \cap D^+_{k+1}$ do not intersect. However, $A'_1 \cup A'_2$ has the same size as A, but the separation between the largest first coordinate of elements of $A'_1 \cup A'_2$ and the smallest first coordinate of $A'_1 \cup A'_2$ is less than $m-t$. Therefore, by induction on $m-t$ we reduce to the case where no σ_r is equal to zero, *i.e.*, there is an element of A with first coordinate r for every $t \leq r \leq m$.

Consider the sets $S_r = \left\{v_{\to j} : v \in A^1_r, j \in \{2,3\}\right\} \subseteq N(A) \cap D^+_{k+1}$. Observe that the cardinality of S_r is at least $\sigma_r + 1$. Clearly all S_r are disjoint, since all elements of S_r have first coordinate r. The elements of S_t have t as their first coordinate, while all elements of B have first coordinates at least $t+1$, so no elements of S_t are in B. Furthermore, for all $r > t$, if an element of S_r is in B, then by considering its first coordinate, the element must be in the set $\left\{v_{\to 1} : v \in A^1_{r-1}\right\}$. In particular, this set has size σ_{r-1}. If $\sigma_r + 1 > \sigma_{r-1}$, then there are at least $\sigma_r + 1 - \sigma_{r-1}$ elements in S_r not in B. Therefore, the number of elements in $N(A) \cap D^+_{k+1}$ that are not in B is bounded below by

$$g(\sigma) := \sum_{r=t}^{m} \max\left(0, \sigma_r + 1 - \sigma_{r-1}\right), \tag{3.9}$$

with the convention that $\sigma_{t-1} = 0$.

Now take any nonzero sequence $\sigma_t, \sigma_{t+1}, \ldots, \sigma_m$. We claim that if $g(\sigma) < f$, then $\sum_{r=t}^{m} \sigma_r \leq \frac{1}{2}(f-1)(f-2)$, which would complete the proof of the theorem. Suppose we have some sequence $\sigma_t, \sigma_{t+1}, \ldots, \sigma_m$ with $g(\sigma) < f$. First, suppose that there exists some $r > t$ where $\sigma_r \geq \sigma_{r-1}$. Then adding 1 to σ_{r-1} decreases the r-th term of (3.9) by 1, possibly adds 1 to the $(r-1)$-st term, and leaves all other terms unchanged; in particular, it does not increase the value of $g(\sigma)$ and increases $\sum \sigma_r$. Therefore, we can reduce to the case where σ is strictly decreasing.

Next, suppose we have $\sigma_r < \sigma_{r-1} - 1$ for some $t < r \leq m$. Then adding 1 to σ_r leaves all terms of (3.9) unchanged. Similar to before, this operation does not change $g(\sigma)$, while increasing $\sum \sigma_r$. Doing this repeatedly, we reduce to the case where

$$\sigma_{r-1} = \sigma_r + 1 \tag{3.10}$$

for all $t < r \leq m$. However, this case is easy to evaluate; each term in (3.9) is zero except the $r = t$ term, which is equal to $\sigma_t + 1$. Since $g(\sigma) = \sigma_t + 1 < f$, $\sigma_t < f - 1$. Since $\sigma_m > 0$,

$\sum_{r=t}^{m} \sigma_r$ is at most the sum of the first $f-2$ positive integers. Thus,

$$\sum_{r=t}^{m} \sigma_r \leq \frac{1}{2}(f-1)(f-2). \qquad \square$$

This allows us to prove the following theorem.

Theorem 3.8. *For any dimension $d \geq 3$ and any fixed positive integer f, f firefighters per time step are not sufficient to contain all finite outbreaks in $\mathbb{L}^d$.*

Proof. Since $\mathbb{L}^3$ is contained in $\mathbb{L}^d$ for $d \geq 3$, it suffices to prove the statement for $d = 3$. We consider an initial outbreak consisting of all of D_k^+ for k large enough so that $\left|D_k^+\right| > \frac{1}{2}(f-1)(f-2)$. To show that f firefighters are insufficient to contain this outbreak, we will construct a related graph that captures the essential disease dynamics and then invoke Theorem 3.2. Let G be the subgraph of $\mathbb{L}^3$ induced by vertices with non-negative coordinates that are distance at least k from the origin. Let G' be the graph formed from G by identifying all of the vertices in D_k^+ as a single vertex r. An edge exists between vertices x and y in G' if xy is an edge in G or if $x = r$ and $y \in N_G(D_k^+)$. Let D_i' denote the set of vertices in G' that are distance i from the root r. By Lemma 3.7,

$$\left|N(D_k^+) \cap D_{k+1}^+\right| \geq \left|D_k^+\right| + f > \frac{1}{2}(f-1)(f-2) + f,$$

and so

$$\left|N(r) \cap D_1'\right| > \left(\left|D_0'\right| - 1\right) + \frac{1}{2}(f-1)(f-2) + f.$$

If $A' \subseteq D_i'$, where $i > 0$ and $|A'| > \frac{1}{2}(f-1)(f-2)$, then A' corresponds to a set $A \subseteq D_{k+i}^+$ and by Lemma 3.7,

$$\left|N(A) \cap D_{k+i+1}^+\right| \geq |A| + f,$$

and hence

$$\left|N(A') \cap D_{i+1}'\right| \geq \left|A'\right| + f.$$

By Theorem 3.2 with $h = 0$, and $a_0 = f$, f firefighters are insufficient to contain an outbreak starting at r in G', and hence f firefighters are insufficient to contain an outbreak consisting of all of D_k^+ in $\mathbb{L}^3$. $\square$

The essential problem here is that for $d \geq 3$, the boundary of an outbreak grows faster than the constant number of firefighters deployed at a given time step. Indeed, in dimension d, the boundary grows as a polynomial of degree $d-2$. This motivates the following ambitious conjecture.

Conjecture 3.9. *Suppose that $f(t)$ is a function on $\mathbb{N}$ with the property that $\frac{f(t)}{t^{d-2}}$ goes to 0 as t gets large. Then there exists some outbreak which cannot be contained by deploying $f(t)$ firefighters at time t.*

A weaker conjecture would require $f(t)$ to be a polynomial.

Lemma 3.7 also allows us to resolve another conjecture of Wang and Moeller in [13]. They had conjectured that as n gets large, the proportion of elements in the three-dimensional grid $P_n \times P_n \times P_n$ which can be saved by using one firefighter per time step when an outbreak at one vertex occurs goes to 0 as n gets large. We prove this conjecture in the following

Theorem 3.10. *Let v be any vertex of $P_n \times P_n \times P_n$, for $n \geq 1$. Then the maximum number of vertices which can be saved by deploying one firefighter per time step with an initial outbreak at v grows at most as $O(n^2)$. In particular, the proportion of vertices which can be saved goes to 0 as n gets large.*

Proof. We prove the theorem in the case $v = (0,0,0)$. The general statement easily follows by splitting $P_n \times P_n \times P_n$ into orthants with apex v. We actually prove a stronger statement. Consider the graph G induced from the lattice $\mathbb{L}^3$ by vertices with non-negative coordinates and distance at most $3n$ from the origin v. We prove the theorem for the graph G. Note that G contains $P_n \times P_n \times P_n$ as an induced subgraph.

We claim that $|B_t| - r_t \geq \frac{t^2+t+2}{2}$ regardless of what firefighter placements are made. Since there are $\binom{t+2}{2} = \frac{t^2+3t+2}{2}$ vertices in D_t^+, this statement is saying that at the end of the t^{th} time step the number of reserve firefighters together with the unburned vertices (including defended vertices) in D_t^+ cannot exceed t. By considering time up to $t = 3n$, when all vertices have had a chance to be burned, at most $1+2+\ldots+3n = O(n^2)$ vertices are unburned. This implies the same conclusion for $P_n \times P_n \times P_n$.

The proof of the claim is by induction. At the end of the 0^{th} time step, there are no reserve firefighters, and one vertex in D_1 is burned; the difference is $1-0=1 \geq 1 = \frac{0^2+0+2}{2}$ as desired.

Suppose $t \geq 0$, and suppose that the statement is true for t. Then

$$|B_t| \geq \frac{t^2+t+2}{2} > \frac{1}{2}t(t+1). \tag{3.11}$$

Let $f = t+2$. By Lemma 3.7,

$$\left|N(B_t) \cap D_{t+1}^{+}\right| \geq |B_t| + f. \tag{3.12}$$

As in the proof of Theorem 3.2, let p_{t+1} denote the number of firefighters placed in D_{t+1}^{+} at time $t+1$, and let $p_{\leq t}$ denote the number of reserve firefighters placed in D_{t+1}^{+} during time steps $1, \ldots, t$. Thus,

$$\begin{aligned}
|B_{t+1}| - r_{t+1} &= \left[\left|N(B_t) \cap D_{t+1}^{+}\right| - p_{t+1} - p_{\leq t}\right] - r_{t+1} \\
&\geq \left|N(B_t) \cap D_{t+1}^{+}\right| - r_t - 1, \text{ by (3.2)}, \\
&\geq |B_t| + f - r_t - 1, \text{ by (3.12)}, \\
&\geq \frac{t^2+t+2}{2} + (t+2) - 1, \text{ by (3.11)}, \\
&\geq \frac{(t+1)^2+(t+1)+2}{2}.
\end{aligned}$$

Hence the claim follows. □

In practice, one can ensure when an outbreak starts at $(0,0,0)$ that t vertices in D_t^{+} are unburned at time t. However, because the fire doubles back on itself, it is unclear that one can actually save a quadratic number of vertices. Wang and Moeller exhibit the construction of building a "fire wall" by defending all of the vertices at distance k from (n,n,n). In order for this to be effective, we must be able to cover all $\frac{(k+1)(k+2)}{2}$ such vertices in the $3n-k$ time steps it takes the fire to reach this hyperplane. This yields $k = O(\sqrt{n})$. The number of vertices saved is the number of vertices at distance k or less from (n,n,n), which is $\frac{(k+1)(k+2)(k+3)}{6}$. This is $O(k^3) = O(n^{3/2})$. Therefore, the optimal number of vertices saved given an initial outbreak at $(0,0,0)$ in the grid graph $P_n \times P_n \times P_n$ when deploying one firefighter per time step is between $O(n^{3/2})$ and $O(n^2)$.

3.2.2 Two Dimensional Square Grid

According to Wang and Moeller in [13], Hartnell, Finbow, and Schmeisser first proved that an outbreak of fire in $\mathbb{L}^2$ starting at a single vertex can be contained using two firefighters per time step. Their sequence of firefighter placements contained the outbreak at the end of 11 time steps. Wang and Moeller showed that the disease cannot be contained at the end of 7 time steps when using two firefighters per time step and presented a sequence of firefighter placements that attains this minimum. Their sequence allows 18 vertices to be burned. Surprisingly, Wang and Moeller do not comment on whether their solution attains the minimum number of burned vertices. In fact, 18 is the minimum number of burned vertices, and we prove this using integer programming. The same technique also gives a computer proof of Wang and Moeller's result that at least 8 time steps are needed. Their proof relies heavily on case analysis.

The tightness in the following theorem is due to Wang and Moeller [13].

Theorem 3.11. *In $\mathbb{L}^2$, if an outbreak of fire starts an a single vertex, then when using two firefighters per time step at least* 18 *vertices are burned. This bound is tight.*

Proof. We formulate an integer program using the boolean variables $b_{x,t}$ and $d_{x,t}$. The variable $b_{x,t}$ is 1 if and only if vertex x is burned at or before time t, and $d_{x,t}$ is 1 if and only if x is defended at or before time t. We wish to minimize the total number of vertices that become burned. For the integer program to be implementable with a finite number of variables and constraints, we restrict the graph to $L = \{(x, y) \in \mathbb{L}^2 : |x| \leq \ell \text{ and } |y| \leq \ell\}$ and $0 \leq t \leq T$, where ℓ and T are chosen to be sufficiently large that the fire never reaches the boundary and is completely contained by time T. In the actual computations performed, $\ell = 6$ and $T = 9$ proved sufficient. We choose $T > 8$ to ensure that the fire is actually contained and does not grow in the last time step.

The integer program is

$$\text{minimize} \sum_{x \in L} b_{x,T}$$

$$\text{subject to:} b_{x,t} + d_{x,t} - b_{y,t-1} \geq 0, \text{ for all } x \in L,\ y \in N(x), \text{ and } 1 \leq t \leq T, \tag{3.13}$$

$$b_{x,t} + d_{x,t} \leq 1, \text{ for all } x \in L \text{ and } 1 \leq t \leq T, \tag{3.14}$$

$$b_{x,t} - b_{x,t-1} \geq 0, \text{ for all } x \in L \text{ and } 1 \leq t \leq T, \tag{3.15}$$

$$d_{x,t} - d_{x,t-1} \geq 0, \text{ for all } x \in L \text{ and } 1 \leq t \leq T, \tag{3.16}$$

$$\sum_{x \in L} (d_{x,t} - d_{x,t-1}) \leq 2, \text{ for } 1 \leq t \leq T, \tag{3.17}$$

$$b_{x,0} = \begin{cases} 1 & \text{if } x \text{ is the origin,} \\ 0 & \text{otherwise,} \end{cases} \quad \text{for all } x \in L, \tag{3.18}$$

$$d_{x,0} = 0, \text{ for all } x \in L, \tag{3.19}$$

$$b_{x,t}, d_{x,t} \in \{0,1\}, \text{ for all } x \in L \text{ and } 0 \leq t \leq T. \tag{3.20}$$

Condition (3.13) enforces the spread of the fire while respecting vertices defended by a firefighter. Note that vertices can spontaneously combust, catching fire, but the minimization of the objective function ensures that this does not happen in the optimal solution. Condition (3.14) prevents a firefighter from defending a burnt vertex, while conditions (3.15) and (3.16) ensure that once a vertex is burnt or defended, it stays in that state. Condition (3.17) only allows two firefighters per time step. Conditions (3.18) and (3.19) give the initial conditions at time $t = 0$, and condition (3.20) makes the program a binary integer program.

The integer program was solved in about 1.83 hours using the GNU Linear Programming Kit [8] running on a Pentium IV 2.6GHz processor, and 18 was the minimum number of burnt vertices at time $t = 9$. Figure 3.1 shows the minimum solution. The fire was completely contained and thus did not reach the sides of L. Also note that the solution presented by Wang and Moeller in [13] also allows only 18 burnt vertices but is slightly different from the solution presented here. □

Lemma 3.12. *If an outbreak of fire in* $\mathbb{L}^2$ *is contained by* 14 *defended vertices and* (x, y)

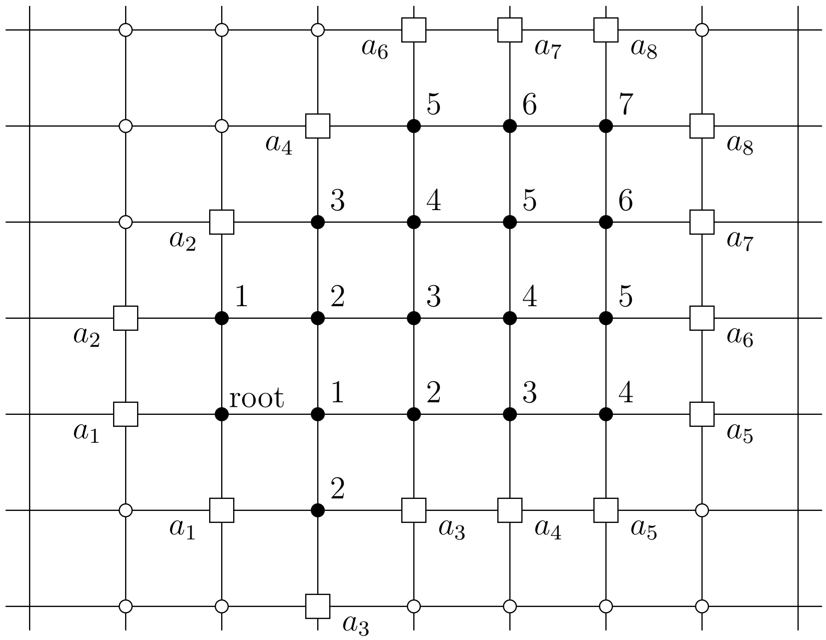

Figure 3.1: Optimal solution of the integer program used in the proof of Theorem 3.11. The fire outbreak starts at time 0 at the root, and then spreads to the black vertices at the times written next to the vertices. The square firefighters a_i are placed at time i. This placement of two firefighters per time step in $\mathbb{L}^2$ completely contains the outbreak in 8 time steps, allowing only the minimum number of 18 burned vertices.

is a burnt vertex, then $|x| \leq 5$ *and* $|y| \leq 5$.

Proof. Suppose that (x, y) is a burnt vertex, and, without loss of generality, that $x > 5$. Since (x, y) is burnt, there is a path $v_0 = (x, y), v_1, v_2, \ldots, v_t = (0, 0)$ from (x, y) to the origin consisting of burnt vertices. For each $0 \leq a \leq 6$, there is a vertex $v_{\rho(a)}$ such that the first coordinate of $v_{\rho(a)}$ is a. Since the fire is contained, there must be a defended vertex above and below each of these seven vertices, and there must be at least one defended vertex with first coordinate less than 0 and one with first coordinate greater than x. But this requires 16 defended vertices, resulting in a contradiction. ☐

Theorem 3.13 (Wang and Moeller). *In* $\mathbb{L}^2$*, if an outbreak of fire starts at a single vertex, then the fire cannot be contained at the end of* 7 *time steps when using two firefighters per time step. Thus, at least* 8 *time steps are needed to contain the fire, and this bound is tight.*

Proof. We use a similar integer program to the one used in the proof of Theorem 3.11. By Lemma 3.12, if the outbreak can be contained after 7 time steps, then no burnt vertex will have either coordinate equaling 6 in absolute value. We thus use the finite grid L where $\ell = 6$, and we use the objective function

$$\text{minimize} \sum_{\substack{x=(a,b)\in L \\ |a|=6 \text{ or } |b|=6}} b_{x,T}.$$

If the disease can be contained after 7 time steps, then the optimal value of the objective function will be 0. All of the conditions from the previous integer program are included except condition (3.17) is changed to

$$\sum_{x \in L} (d_{x,t} - d_{x,t-1}) \leq \begin{cases} 2 & \text{for } 1 \leq t \leq 7, \\ 0 & \text{for } 8 \leq t \leq T. \end{cases} \tag{3.21}$$

This prevents firefighters from being used after 7 time steps.

The integer program with $T = 9$ was solved in about 40 minutes using the GNU Linear Programming Kit running on a Pentium M 900MHz processor. The minimum value was 1, meaning that in every feasible solution, the fire burned a vertex with one coordinate equaling 6 in absolute value. This contradicts Lemma 3.12, and so at least 8 time steps are needed to contain an outbreak in $\mathbb{L}^2$ when using two firefighters per time step. □

3.3 NP-Completeness

MacGillivray and Wang [11] formulated the problem of finding the optimal placement of firefighters as a decision problem and showed that the problem is NP-complete. While straightforward, their construction does have a large number of vertices and an average degree that asymptotically is four. We present here an alternate proof that is a reduction to the satisfiability problem SAT. Our construction uses fewer vertices and the average degree asymptotically is two. In some models of disease spread, a low average degree is more realistic. For instance, in models of sexually transmitted diseases, most people have few sexual partners in a given period of time such as a week. In these instances, our construction is more appropriate. We show average degree calculations after we present our construction.

Definition 3.14. Let FIREFIGHTER be the following decision problem:

Instance: A finite rooted graph (G, r) and an integer $p \geq 1$.

Question: Is there a finite sequence $a_1, a_2, \ldots, a_t$ of vertices of G such that if an outbreak of fire starts at the root r at time 0 and vertex a_i is defended at time i, then

1. Vertex a_i is neither burning nor defended at the beginning of the i^{th} time step and hence can be defended at time i.

2. There is no undefended unburnt vertex adjacent to a burning vertex at the end of the t^{th} time step.

3. At least p vertices are saved at the end of the t^{th} time step.

Note that only one firefighter is deployed per time step.

Theorem 3.15 (MacGillivray and Wang). FIREFIGHTER *is* NP-*complete.*

To explain our construction, we first think of a binary tree (not necessarily complete),

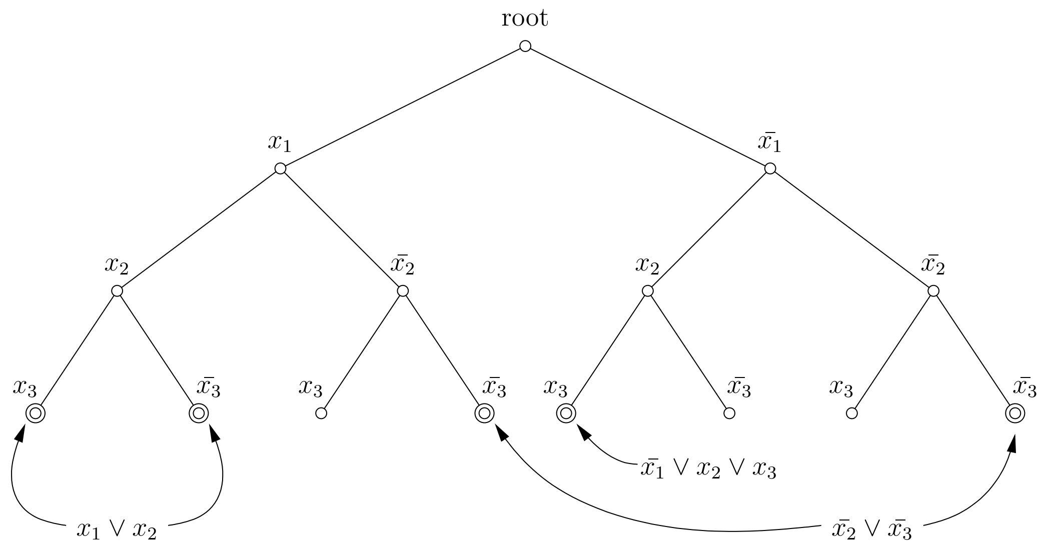

Figure 3.2: Reduction of the formula $\varphi = (x_1 \vee x_2) \wedge (\bar{x_1} \vee x_2 \vee x_3) \wedge (\bar{x_2} \vee \bar{x_3})$ to a binary tree.

where the root of the tree is where the fire outbreak begins. Each level of the tree is associated with a boolean variable x_i, and each leaf represents a disjunctive clause. Figure 3.2 shows the construction of a tree for the formula $\varphi = (x_1 \vee x_2) \wedge (\bar{x_1} \vee x_2 \vee x_3) \wedge (\bar{x_2} \vee \bar{x_3})$. Recall that for trees there is a sequence of vertices $a_1, a_2, \ldots$ attaining the maximum number of saved vertices where vertex a_i is on level i. The firefighter placements correspond to a truth assignment for φ in a natural way: if a_1 is the left vertex, then x_1 is true, otherwise x_1 is false; and so on for each a_i and x_i. If a leaf vertex is saved, then some ancestor (or itself) was defended, indicating that some literal in the corresponding clause is set to true. Thus, the formula evaluates to false if and only if the fire reaches one of the leaves corresponding to a clause in φ. We add h pendant vertices to each clause vertex, so that if a clause vertex is burned then at least $h-1$ other vertices are as well. We call such a vertex with h pendant vertices a "super-spreader vertex." Thus, when h is very large, there exists a truth assignment satisfying φ if and only if there is a firefighter sequence that saves all of the vertices except at most h.

Clearly this construction is a reduction of SAT to FIREFIGHTER, but it is not a polynomial reduction. The difficulty can be seen in Figure 3.2: several leaf vertices may be associated to the same clause. To remedy this difficulty, we introduce a more complicated construction, but the proof idea is still the same.

Proof of Theorem 3.15. FIREFIGHTER is in NP since it can be verified in polynomial time whether a given sequence of firefighter placements saves p vertices. We show that FIREFIGHTER is NP-hard by reducing SAT to FIREFIGHTER. Let

$$\varphi = C_1 \wedge C_2 \wedge \ldots \wedge C_\ell = (c_{1,1} \vee c_{1,2} \vee \ldots \vee c_{1,k_1}) \wedge (c_{2,1} \vee c_{2,2} \vee \ldots \vee c_{2,k_2}) \wedge \ldots \wedge (c_{\ell,1} \vee c_{\ell,2} \vee \ldots \vee c_{\ell,k_\ell})$$

be a boolean formula in conjunctive normal form over the k variables $x_1, x_2, \ldots, x_k$. Let $\widetilde{\varphi} = \varphi \wedge x_1 \wedge x_2 \wedge \ldots \wedge x_k \wedge \bar{x_1} \wedge \bar{x_2} \wedge \ldots \wedge \bar{x_k}$. If φ already contains any of the singleton literals, then the literal is not repeated.

Construct a rooted ternary tree T_1 (not necessarily complete) of height k where each vertex on level i encodes a clause D_v that contains at most the variables $x_1, \ldots, x_i$. The clause D_v can be empty. Define T_1 inductively as follows:

Level $i = 0$: Place a single root vertex on level 0. This vertex encodes the empty clause.

To define level i from level $i-1$: We say that a clause C of $\widetilde{\varphi}$ is *compatible* with D_v if every truth assignment τ of $x_1, \ldots, x_k$ that satisfies C also satisfies D_v. For every vertex v on level $i-1$ such that there exists a clause C of $\widetilde{\varphi}$ compatible with D_v, add three new children of v to level i, where the first child encodes $D_v \vee x_k$, the second child D_v, and the third child $D_v \vee \bar{x_k}$.

We call each of the vertices on level k that encodes a clause of φ a "clause vertex." We will also sometimes refer to a "clause vertex of $\widetilde{\varphi}$" when a vertex on level k encodes a clause of $\widetilde{\varphi}$. Figure 3.3 shows the tree T_1 for $\varphi = (x_1 \vee x_2) \wedge (\bar{x_1} \vee x_2 \vee x_3) \wedge (\bar{x_2} \vee \bar{x_3})$.

We construct a new tree T_2 from T_1 by subdividing edges. For $1 \leq i \leq k-1$, denote by L_i the vertices of T_1 on level i that have a child. Let $L_i = \{v_1, v_2, \ldots, v_{q_i}\}$. We perform the following operation for each $1 \leq i \leq k-1$ in order: for $1 \leq r \leq q_i$, add $r-1$ vertices to the edge above v_r leading to the root and $q_i - r$ vertices to the edges below v_r. Observe that after the subdivisions are performed, the tree T_2 is of height $\sum_{i=1}^{k-1}(q_i+1)+1$ and vertex $v_r \in L_i$ is on level $\sum_{j=1}^{i-1}(q_i+1)+r-1$ in T_2. Thus, except for levels 0 and $\sum_{i=1}^{k-1}(q_i+1)+1$ of T_2, there is exactly one vertex from $L_1 \cup \ldots \cup L_{k-1}$ on each level of T_2. Figure 3.4 shows

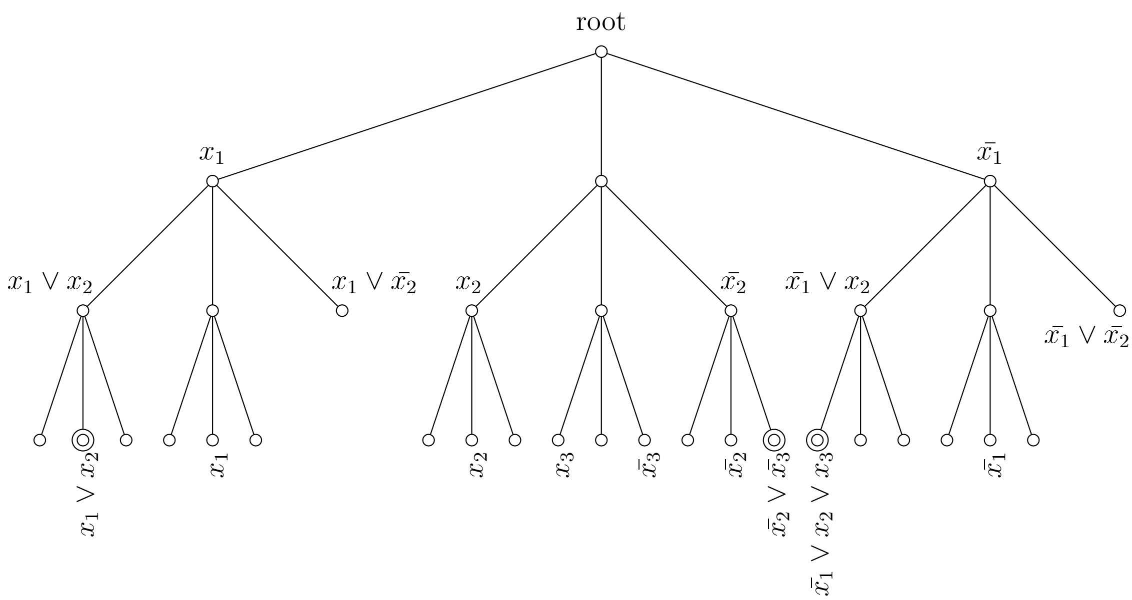

Figure 3.3: Construction of the tree T_1 for $\varphi = (x_1 \vee x_2) \wedge (\bar{x_1} \vee x_2 \vee x_3) \wedge (\bar{x_2} \vee \bar{x_3})$. Clause vertices of φ are marked with double circles, and clause vertices of $\widetilde{\varphi}$ are labeled. Vertices on level 1 that contain x_1 or $\bar{x_1}$ and vertices on level 2 that contain x_2 or $\bar{x_2}$ are also labeled.

the tree T_2 for $\varphi = (x_1 \vee x_2) \wedge (\bar{x_1} \vee x_2 \vee x_3) \wedge (\bar{x_2} \vee \bar{x_3})$.

Form a new tree T_3 by subdividing every edge once. Figure 3.5 shows the tree T_3 for $\varphi = (x_1 \vee x_2) \wedge (\bar{x_1} \vee x_2 \vee x_3) \wedge (\bar{x_2} \vee \bar{x_3})$. For $1 \leq i \leq k-1$, let w_i be the vertex of L_i that is on the lowest level of T_3. Let W_i be the set of all vertices in T_3 that are on the same level as w_i. Form a new tree T_4 by subdividing each edge once below a vertex in W_i, for $1 \leq i \leq k-1$. Figure 3.6 shows the tree T_4 for $\varphi = (x_1 \vee x_2) \wedge (\bar{x_1} \vee x_2 \vee x_3) \wedge (\bar{x_2} \vee \bar{x_3})$.

We are now going to add cycles to our construction, and hence it will no longer be a tree. For every even level of T_4 except the bottom level there is exactly one vertex from T_1. There are no vertices from T_1 on the odd levels. Every vertex v from T_1 encodes some clause D_v. We extend this encoding to T_4 where if a vertex u in T_4 is not in T_1, then u encodes the clause D_v for its closest descendant v from T. Note that v is unambiguously defined since the only vertices in T_4 with more than one child are from T_1. Form a new graph G_5 from T_4 by replacing every vertex v from T_1, its three children, and its three grandchildren with the decision widget shown in Figure 3.7. Note that the vertices v, a, b, and c shown in T_4 on the left correspond to the vertices with the same labels shown in G_5 on the right,

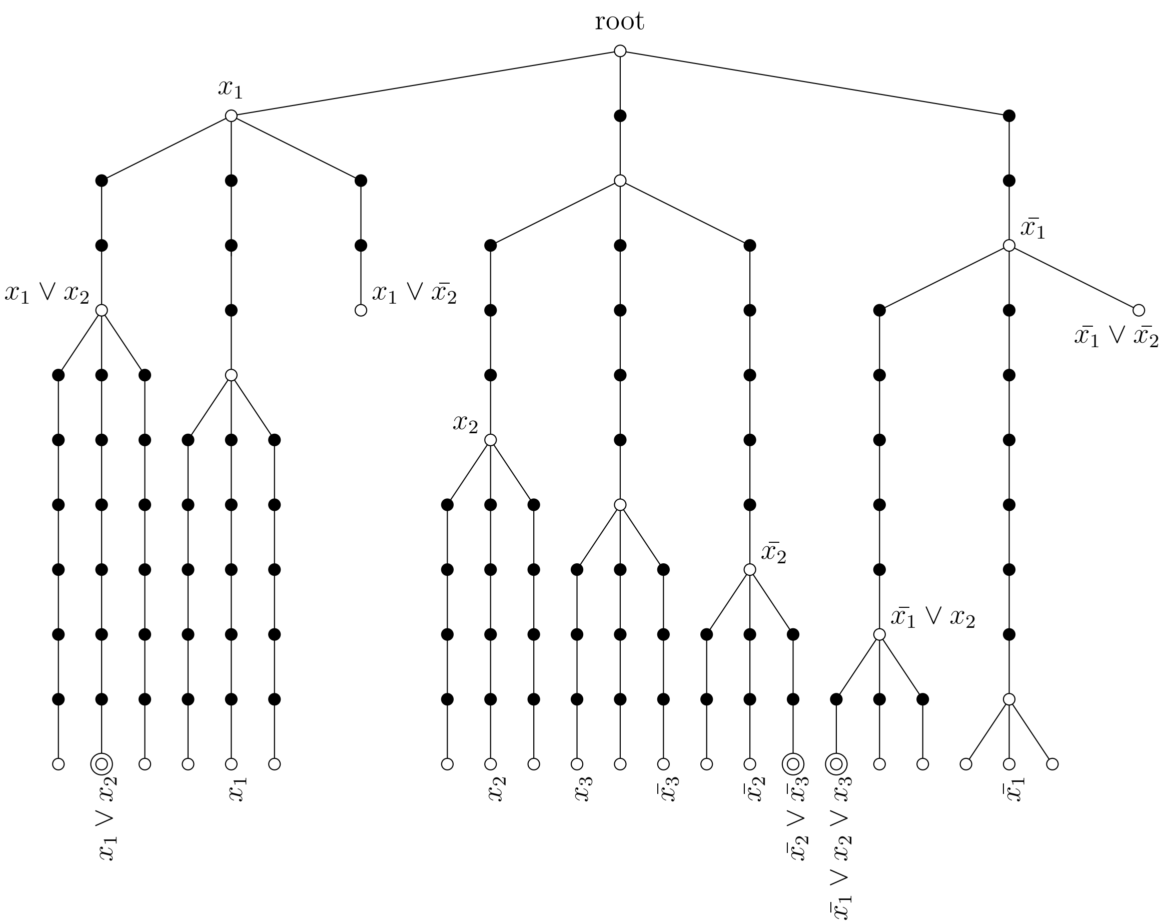

Figure 3.4: Construction of the tree T_2 for $\varphi = (x_1 \vee x_2) \wedge (\bar{x_1} \vee x_2 \vee x_3) \wedge (\bar{x_2} \vee \bar{x_3})$. Vertices that are also in T_1 are marked with hollow dots, and new vertices are marked with black dots.

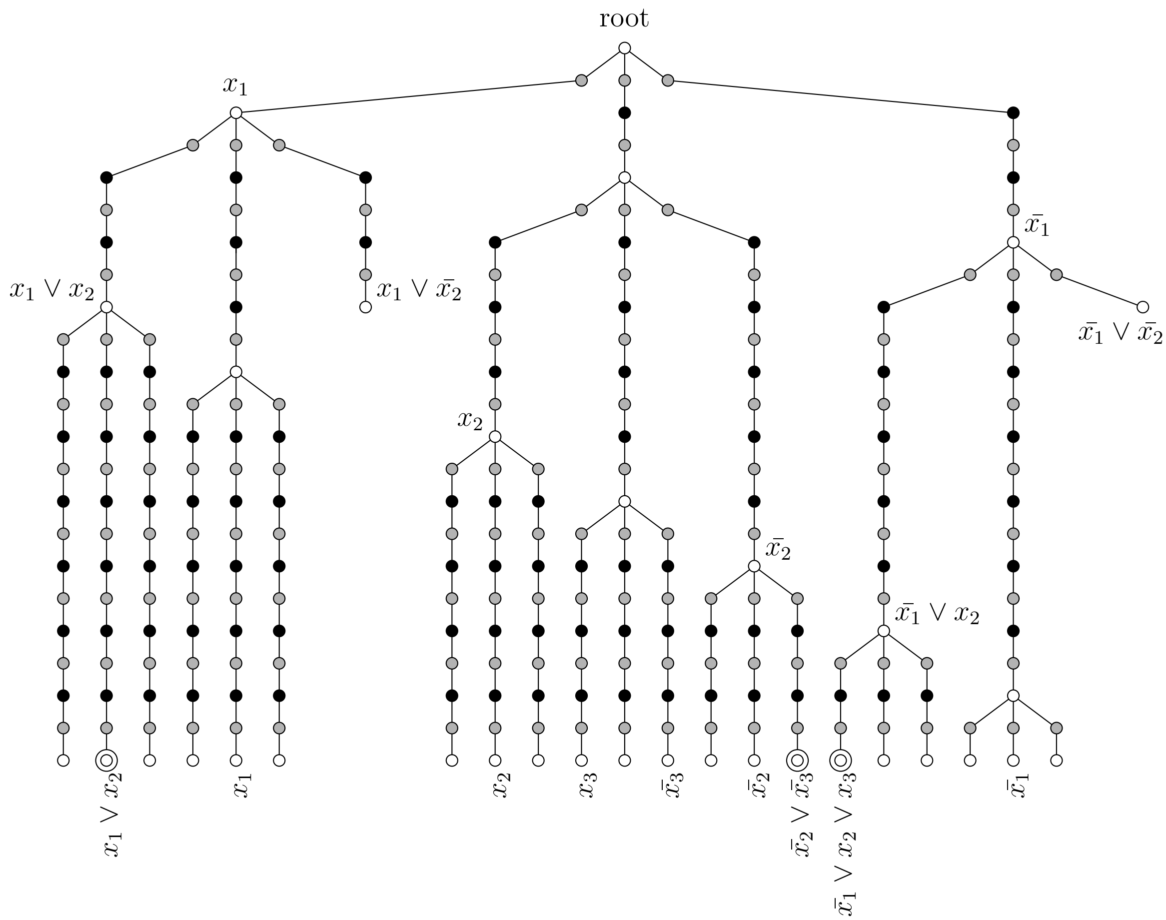

Figure 3.5: Construction of the tree T_3 for $\varphi = (x_1 \vee x_2) \wedge (\bar{x_1} \vee x_2 \vee x_3) \wedge (\bar{x_2} \vee \bar{x_3})$. Vertices that are also in T_1 are marked with hollow dots, and new vertices on subdivided edges are marked with gray dots.

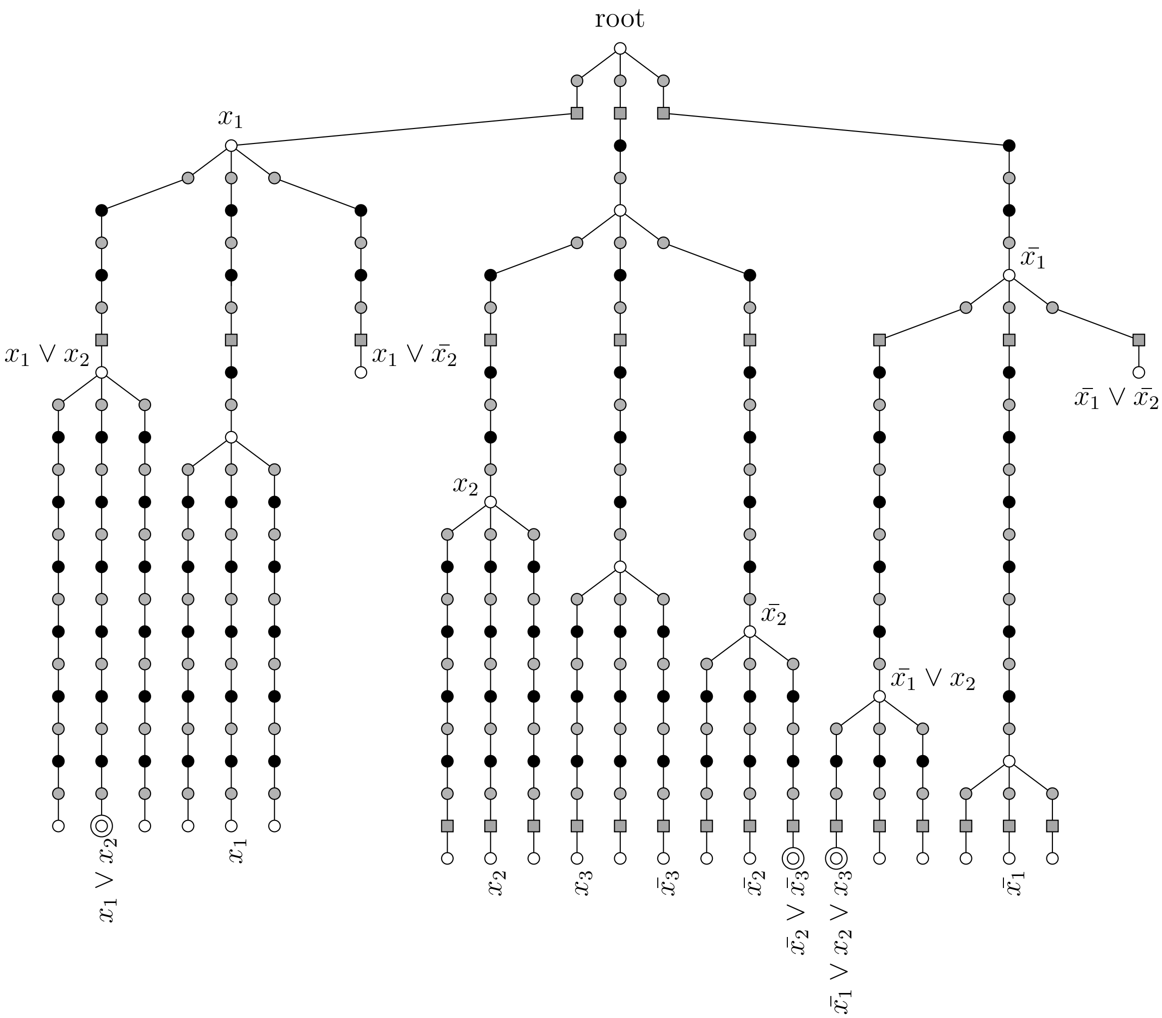

Figure 3.6: Construction of the tree T_4 for $\varphi = (x_1 \vee x_2) \wedge (\bar{x_1} \vee x_2 \vee x_3) \wedge (\bar{x_2} \vee \bar{x_3})$. Vertices that are also in T_1 are marked with hollow dots, vertices added when forming T_3 are marked by gray dots, and those added when forming T_4 are marked by gray squares.

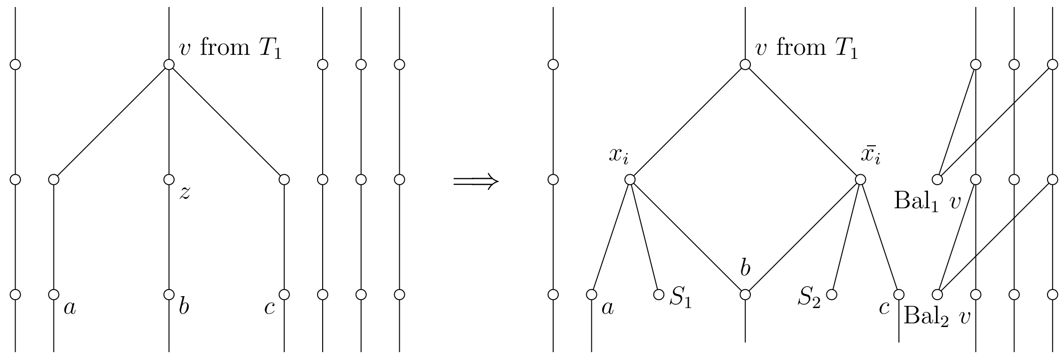

Figure 3.7: The decision widget for x_i. The vertices v, a, b, and c shown in T_4 on the left correspond to the vertices with the same labels shown in G_5 on the right. The vertex labeled z is removed when forming G_5.

and that the vertex labeled z is removed when forming G_5. We say the decision widget "encodes the truth value for x_i" if v is on level i in T_1. The vertices marked S_1 and S_2 are "super-spreader" vertices to which we will attach h pendant vertices. The value of h will be specified below. Additionally, if D_v does not encode the empty clause, create two new vertices called Bal$_1$ v and Bal$_2$ v, where Bal$_1$ v is on the level below v and Bal$_2$ v is two levels below. Connect Bal$_1$ v to all of the vertices on the level of v that encode a singleton clause that is the negation of some literal appearing in D_v. Similarly connect Bal$_2$ v to all of the vertices on the level below v that encode a singleton clause that is the negation of some literal appearing in D_v. For instance, if $D_v = x_1 \vee \bar{x_3}$, then Bal$_1$ v is connected to vertices that encode $\bar{x_1}$ and x_3. Such vertices must exist since the singleton clauses were added to $\widetilde{\varphi}$. Both Bal$_1$ v and Bal$_2$ v are also super-spreader vertices. Figure 3.8 shows the graph G_5 for the formula $\varphi = (x_1 \vee x_2) \wedge (\bar{x_1} \vee x_2 \vee x_3) \wedge (\bar{x_2} \vee \bar{x_3})$.

For each $1 \leq i \leq k-1$, form a new graph G_6 from G_5 by adding two new vertices called Sync x_i and Sync $\bar{x_i}$ on the level below w_i. Connect Sync x_i to each vertex u on the level of w_i where the encoding D_u contains x_i, and connect Sync $\bar{x_i}$ to each vertex u on the level of w_i where the encoding D_u contains $\bar{x_i}$. Both Sync x_i and Sync $\bar{x_i}$ are also super-spreader vertices. Figure 3.9 shows the graph G_6 for the formula $\varphi = (x_1 \vee x_2) \wedge (\bar{x_1} \vee x_2 \vee x_3) \wedge (\bar{x_2} \vee \bar{x_3})$.

Mark all of the clause vertices from T_1 as super-spreader vertices. Form a new graph G_7 from G_6 by adding h pendant vertices to each super-spreader vertex. This finishes the

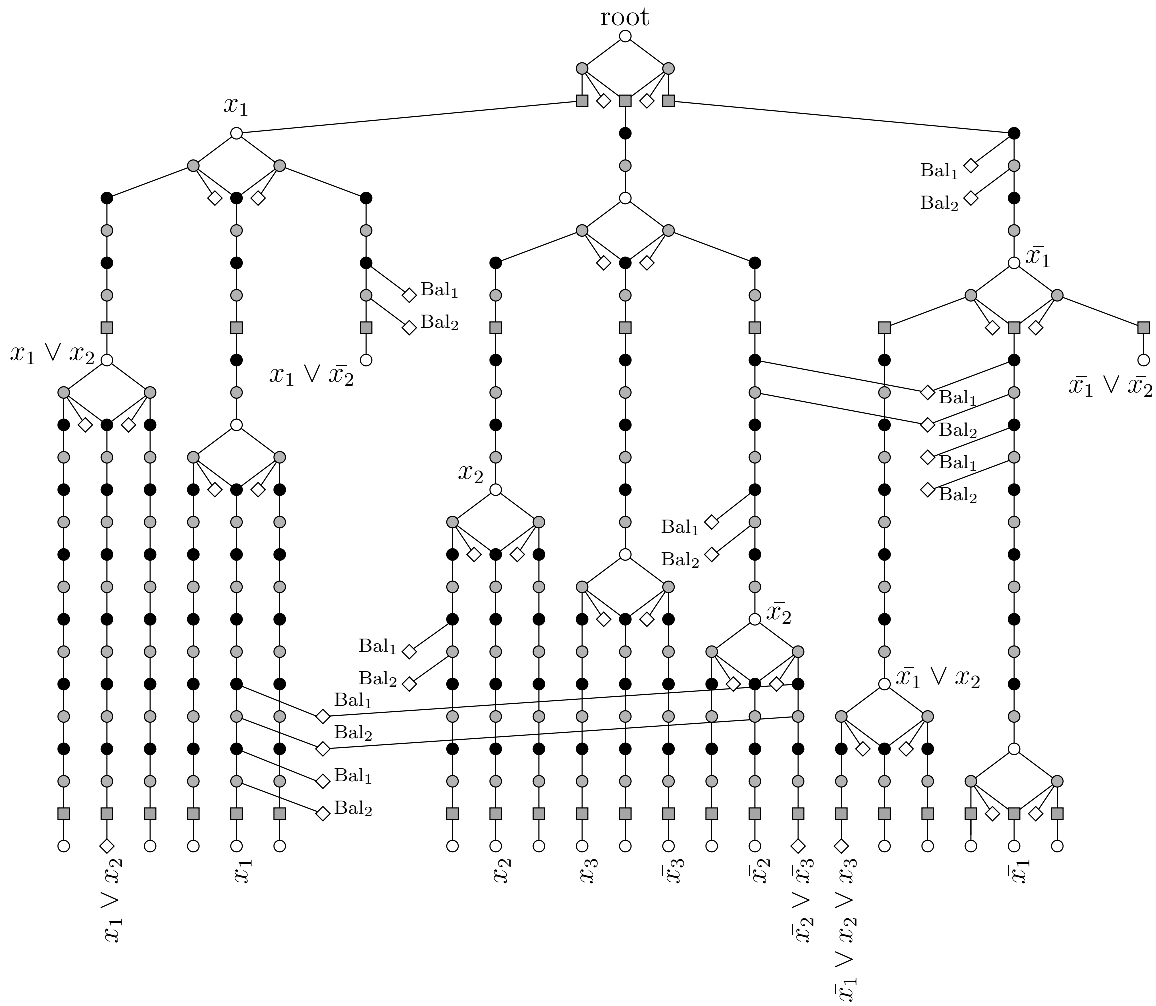

Figure 3.8: Construction of the graph G_5 for $\varphi = (x_1 \vee x_2) \wedge (\bar{x_1} \vee x_2 \vee x_3) \wedge (\bar{x_2} \vee \bar{x_3})$. Vertices added when forming T_3 are marked by gray dots, and those added when forming T_4 are marked by gray squares. Super-spreader vertices are marked with a hollow diamond.

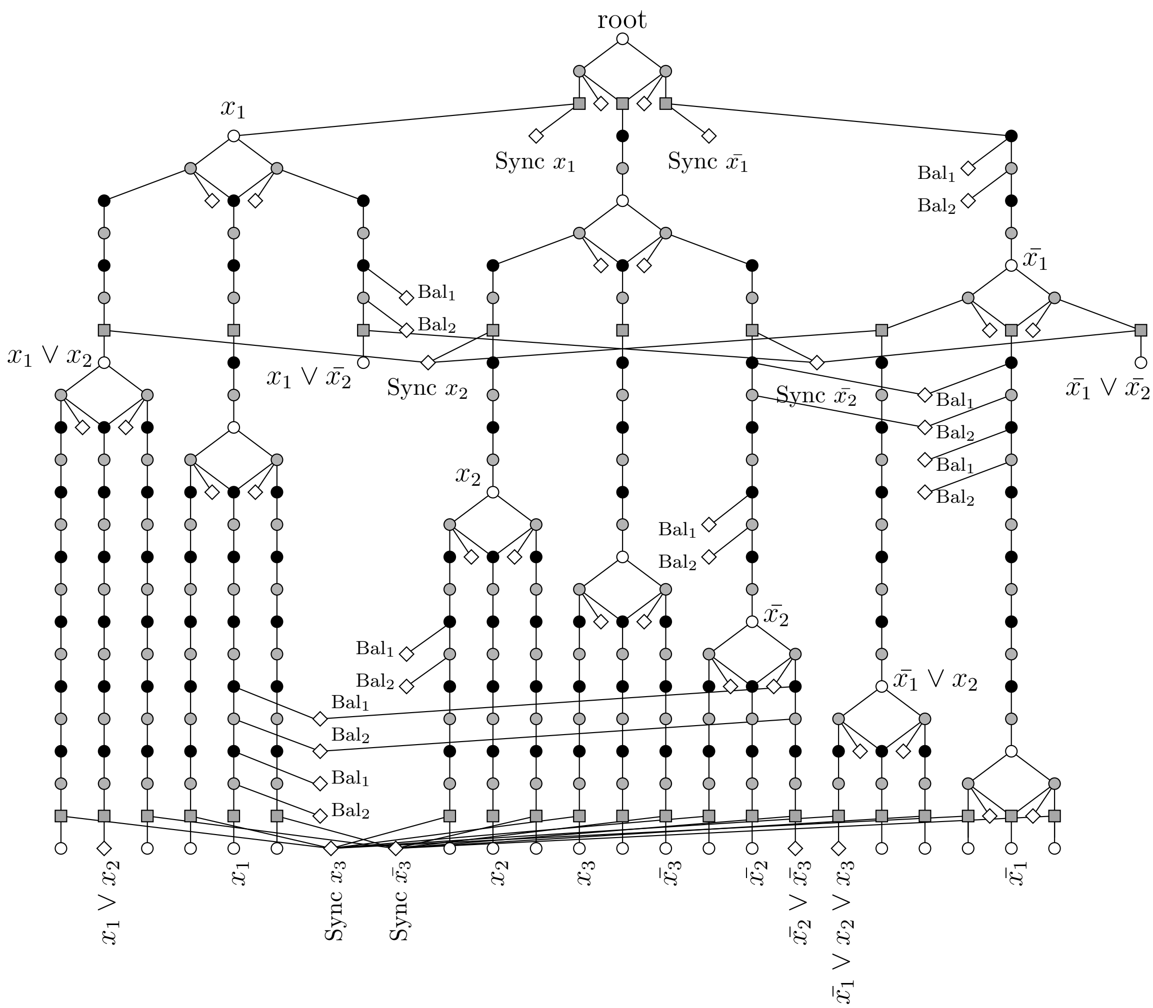

Figure 3.9: Construction of the graph G_6 for $\varphi = (x_1 \vee x_2) \wedge (\bar{x_1} \vee x_2 \vee x_3) \wedge (\bar{x_2} \vee \bar{x_3})$. Vertices added when forming T_3 are marked by gray dots, and those added when forming T_4 are marked by gray squares. Super-spreader vertices are marked with a hollow diamond.

construction of the graph.

To see that the construction is polynomial in size, we bound the number of vertices present in various stages of the construction. The formula $\widetilde{\varphi}$ has at most $\ell + 2k$ clauses, and so level k of T_1 has at most $3(\ell + 2k)$ vertices since every vertex's parent has a clause vertex of $\widetilde{\varphi}$ as a descendant. Thus, T_1 has at most $(k+1) \cdot 3(\ell + 2k)$ vertices, since there are $k+1$ levels.

T_2 also has at most $3(\ell + 2k)$ vertices on each level, and the height of the tree is $\sum_{i=1}^{k-1}(q_i + 1) + 1$. Note that $q_i \leq 3(\ell + 2k)$ since every level on T_1 has at most $3(\ell + 2k)$ vertices. Thus, T_2 has height at most $3k(\ell + 2k + 1)$ and at most $9k(\ell + 2k + 1)^2$ vertices. T_3 has at most twice the number of vertices of T_2, and T_4 adds at most $k \cdot 3(\ell + 2k)$ vertices. Thus, T_4 has at most $18k(\ell + 2k + 1)^2 + 3k(\ell + 2k)$ vertices.

For each vertex v of T_1 replaced by a decision widget, at most three new vertices are added to T_4 to form G_5. Thus,

$$\begin{aligned}
|V(G_5)| &\leq 18k(\ell + 2k + 1)^2 + 3k(\ell + 2k) + 3\,|V(T_1)| \\
&= 18k(\ell + 2k + 1)^2 + 3k(\ell + 2k) + 9(k+1)(\ell + 2k) \\
&\leq 18k(\ell + 2k + 1)^2 + (12k + 9)(\ell + 2k) \\
&\leq 18k(\ell + 2k + 1)^2 + 15k(\ell + 2k) \\
&\leq 33k(\ell + 2k + 1)^2
\end{aligned}$$

vertices, assuming that $k \geq 3$ and $\ell \geq 2$. To form G_6 we add $2k$ vertices, and so

$$\begin{aligned}
|V(G_6)| &\leq 33k(\ell + 2k + 1)^2 + 2k \\
&\leq 34k(\ell + 2k + 1)^2.
\end{aligned}$$

The number s of super-spreader vertices in G_6 is bounded by

$$\begin{aligned}
s &\leq 4\,|V(T_1)| + (\text{number of Sync vertices}) + (\text{number of clause vertices}) \\
&\leq 12(k+1)(\ell + 2k) + 2k + \ell \\
&\leq 17k(\ell + 2k).
\end{aligned}$$

By choosing $h = 3\,|V(G_6)|$, the number of vertices in G_7 is

$$\begin{aligned}|V(G_7)| &= sh + |V(G_6)| \\ &= |V(G_6)|\,(3s+1) \\ &\le 34k(\ell+2k+1)^2\,[51k(\ell+2k)+1] \\ &\le 34k(\ell+2k+1)^2\,[51k(\ell+2k+1)] \\ &= 1734k^2(\ell+2k+1)^3,\end{aligned}$$

which is clearly polynomial in k and ℓ.

Our instance of FIREFIGHTER is G_7, the root from T_1, and $p = |V(G_7)| - h/2$. Observe that a sequence $a_1, a_2, \ldots, a_t$ of firefighter placements saves $|V(G_7)| - h/2$ vertices if and only if no super-spreader vertex becomes burned, since then $h-1$ of the pendant children will also be burned.

We now wish to show that our construction is a reduction from SAT to FIREFIGHTER. Suppose that $a_1, a_2, \ldots, a_t$ is a sequence of firefighter placements that saves at least $n - h/2$ vertices. We show that this sequence gives rise to a truth assignment τ that satisfies φ by proving the following four claims.

Claim 3.16. Vertex a_j is on level j.

Claim 3.17. The decision widgets that encode the truth value for x_i are synchronized in the sense that either the x_i vertex is defended by a firefighter in all of the decision widgets that encode the truth value for x_i in which firefighters are placed, or $\bar{x}_i$ is defended in all of these widgets. This choice defines the truth value for x_i in the truth assignment τ by taking x_i to be true if it is defended and by taking x_i to be false if $\bar{x}_i$ is defended.

Claim 3.18. Every vertex v below the level of Sync x_i is saved if the clause encoded by v is satisfied by the truth assignment τ restricted to the first i variables. If a vertex v is below the level of Sync x_i and no x_{i+1} or $\bar{x}_{i+1}$ from a decision widget for x_{i+1} appears on a shortest path between v and the root, then v is burned if the clause encoded by v is not satisfied by τ restricted to the first i variables.

Claim 3.19. The synchronization vertex Sync x_i is defended if x_i is false in τ and Sync $\bar{x_i}$ is defended if x_i is true in τ.

Proof of claims. Let j_i be the index of the level on which Sync x_i appears, and define j_0 to be 0. Note that all decision widgets for x_i appear on levels between levels j_{i-1} and j_i. We now proceed to prove the claims by induction on i. Suppose that $i = 1$. Then between levels $j_0 + 1$ and j_1 inclusive, there are four super-spreader vertices: S_1 and S_2 in the decision widget that encodes the truth value for x_1 and the two synchronization vertices Sync x_1 and Sync $\bar{x_1}$. Since no super-spreader vertex may be burnt, a_1 must be either the vertex labeled x_1 or $\bar{x_1}$, a_2 the opposite super-spreader vertex in the decision widget for x_1, and a_3 the opposite synchronization vertex. Thus, all four claims are satisfied for $i = 1$ and levels $j \leq j_1$.

Now suppose that $i > 1$ and the claims hold for levels less than or equal to j_{i-1} and decision widgets that encode the truth value for variables with index less than i. By Claim 3.17, the firefighter sequence $a_1, a_2, \ldots, a_{j_{i-1}}$ defines a truth assignment τ_{i-1} for $x_1, x_2, \ldots, x_{i-1}$. If there is a decision widget that encodes the truth value for x_i whose vertices have not already been saved, we call the widget "under active consideration." Note that on each level j where $j_{i-1} < j < j_i$, there are exactly two vertices from the lower two levels of a decision widget that encodes the truth value for x_i. Suppose that v is the vertex at the top of the decision widget on level $j - 1$. If v does not encode the empty clause, then there exist two balance vertices, $\text{Bal}_1\ v$ on level j and $\text{Bal}_2\ v$ on level $j + 1$. This follows immediately from the construction. If v is not burned at the end of $j - 1$ time steps, then at least one of the vertices to which $\text{Bal}_1\ v$ are connected is not satisfied by the truth assignment τ_{i-1} and hence by Claim 3.18 is burned at time $j - 1$. Thus, the one firefighter available at time j must be used to defend $\text{Bal}_1\ v$ on level j, and the one firefighter available at time $j + 1$ must be used to defend $\text{Bal}_2\ v$ on level $j + 1$. Suppose that v is burned at the end of $j - 1$ time steps. Note that if v encodes the empty clause, then the truth assignment τ_{i-1} never satisfies the empty clause, and hence by Claim 3.18 is burned. Since v is burned, then, by construction, all of the vertices to which $\text{Bal}_1\ v$ are connected are satisfied by the truth assignment τ_{i-1} and hence by Claim 3.18 are saved. In order for S_1 and S_2 to be saved on level $j+1$, there are only four possibilities for a_j and a_{j+1}:

$$a_j = x_i, \quad a_{j+1} = S_2, \text{ or}$$
$$a_j = \bar{x}_i, \quad a_{j+1} = S_1, \text{ or}$$
$$a_j = S_1, \quad a_{j+1} = S_2, \text{ or}$$
$$a_j = S_2, \quad a_{j+1} = S_1.$$

Note that two firefighters are used for levels j and $j+1$, and so there are no extra firefighters. If either of the last two options is chosen, then fire will spread to vertices that have x_i in their encodings and to vertices that have $\bar{x}_i$ in their encodings. Thus, both Sync x_i and Sync $\bar{x}_i$ will be threatened with fire at the end of $j_i - 1$ time steps, and only one of the synchronization vertices can be saved by the one firefighter available at time $j_i - 1$. Thus, the last two options for a_j and a_{j+1} are not possible.

Now suppose that two different choices of x_i and $\bar{x}_i$ are made in two decision widgets that encode the truth value for x_i. Then both Sync x_i and Sync $\bar{x}_i$ will be threatened with fire at the end of j_i-1 time steps, and only one of the synchronization vertices can be saved by the one firefighter available at time $j_i - 1$. Thus all of the decision widgets that encode the truth value for x_i are synchronized, and by the construction, every vertex below the level of Sync x_i is saved if satisfied by τ_i. Thus all of the claims are established for levels j where $j \le j_i$ and decision widgets that encode the truth value for variables with index less than or equal to i. □

By Claim 3.17 the sequence $a_1, a_2, \ldots, a_t$ of firefighter placements defines a truth assignment τ for the variables $x_1, \ldots, x_k$, and since the fire reaches no clause vertex, every clause vertex is satisfied by τ. Hence, τ satisfies φ.

For the converse, we construct a sequence $a_1, a_2, \ldots, a_t$ of firefighter placements given a truth assignment τ. As before, let j_i be the index of the level on which Sync x_i appears, and define j_0 to be 0. We construct the sequence iteratively. If x_1 is true in τ, then set $a_1 = x_1$, $a_2 = S_2$, and $a_3 =$ Sync x_i. If x_1 is false in τ, then set $a_1 = \bar{x}_1$, $a_2 = S_1$, and $a_3 =$ Sync $\bar{x}_i$. Recall that on each level j where $j_{i-1} < j < j_i$, there are exactly two vertices from the lower two levels of a decision widget that encodes the truth value for x_i. Suppose that v is the vertex at the top of the decision widget on level $j-1$. By Claim 3.18, either the vertices x_i and $\bar{x_i}$ on level j are saved or $\mathrm{Bal}_1\ v$ on level j is saved. Similarly, either the vertices S_1 and S_2 on level $j+1$ are saved or $\mathrm{Bal}_2\ v$ on level $j+1$ is saved. If x_i and $\bar{x_i}$ on level j are not saved, then choose a_j to be x_i if x_i is true in τ or choose a_j to be $\bar{x_i}$ if x_i is false in τ. If $\mathrm{Bal}_1\ v$ is not saved, then choose a_j to be $\mathrm{Bal}_1\ v$. If S_1 and S_2 on level $j+1$ are not saved, then choose a_{j+1} to be S_2 if x_i is true in τ or choose a_{j+1} to be S_1 if x_i is false in τ. If $\mathrm{Bal}_2\ v$ is not saved, then choose a_{j+1} to be $\mathrm{Bal}_2\ v$. For level j_i, choose a_{j_i} to be Sync $\bar{x_i}$ if x_i is true in τ or choose a_{j_i} to be Sync x_i if x_i is false in τ. By Claim 3.18, the fire reaches a clause vertex v only if no vertex in a decision widget is defended on the shortest path from v to the root. However, if τ satisfies φ, then for each clause vertex v, there is some x_i or $\bar{x_i}$ vertex on the shortest path from v to the root that is defended. Here, x_i or $\bar{x_i}$ is some variable appearing in the clause D_v encoded by v that is set to true by τ. Hence, if τ satisfies φ, then the sequence $a_1, a_2, \ldots, a_t$ saves all of the super-spreader vertices, including the clause vertices, and so saves $|V(G_7)| - h/2$ vertices. □, Theorem 3.15.

3.3.1 Number of Vertices and Average Degree Comparison

In MacGillivray and Wang's proof of the NP-completeness of FIREFIGHTER, their construction has a large number of vertices and an average degree that asymptotically is four. The construction used in our proof has fewer vertices and the average degree asymptotically is two, which for some models is more realistic. We show here the calculations of these parameters.

MacGillivray and Wang proved that FIREFIGHTER is NP-complete by reducing from the problem Exact Cover with 3-sets. In this problem, a set X and a collection $\mathcal{C}$ of 3-subsets of X are given, where $|X| = 3q$ and $|\mathcal{C}| = c$, and the question is whether a subcollection of $\mathcal{C}$ of size q exists that exactly covers X. Let d be the number of disjoint pairs in $\mathcal{C}$. Then MacGillivray and Wang's construction of a graph from an instance of this problem has n_1

vertices and e_1 edges, where

$$n_1 = 1 + cq + 10q^5 d$$

$$e_1 = cq + 2 \cdot 10q^5 d$$

$$\text{average degree} = \frac{2e_1}{n_1} \to 4 \text{ as } q \to \infty.$$

In our proof, we reduce SAT to FIREFIGHTER. If φ is a boolean formula in conjunctive normal form with k variables and ℓ clauses, then our construction has n_2 vertices, where

$$n_2 \leq 1734k^2(\ell + 2k + 1)^3.$$

To calculate the average degree, we divide the vertices of the graph into super-spreader vertices, pendants of super-spreader vertices, and other vertices. Note that all vertices except super-spreader vertices have degree at most four, and super-spreader vertices are connected to h pendant vertices and at most $2k$ other vertices. There are s super-spreader vertices, h pendant vertices attached to each super-spreader vertex, and $|V(G_6)| - s$ other vertices, so

$$\begin{aligned}
\text{average degree} &\leq (h + 2k)\frac{s}{n_2} + 1\frac{sh}{n_2} + 4\frac{|V(G_6)| - s}{n_2} \\
&= \frac{2sh + 2ks + 4\left(|V(G_6)| - s\right)}{n_2} \\
&= 2\frac{sh + |V(G_6)|}{n_2} + \frac{2ks - 4s + 2\,|V(G_6)|}{n_2} \\
&= 2 + \frac{2ks - 4s + 2\,|V(G_6)|}{(3s+1)\,|V(G_6)|}, \text{ since } h = 3\,|V(G_6)|.
\end{aligned}$$

Note that $s > k$ and $s > \ell$. Observe that $|V(G_6)| \geq ck^2$ for some constant $c > 0$ by considering the singleton literals added to $\widetilde{\varphi}$. Thus,

$$\frac{2ks - 4s + 2\,|V(G_6)|}{(3s+1)\,|V(G_6)|} \to 0$$

as $k \to \infty$ and $\ell \to \infty$, and so the the average degree of our construction tends to 2 as $k \to \infty$ and $\ell \to \infty$.

3.4 Miscellaneous Results and Future Work

In this chapter we present some miscellaneous results about trees and some directions for future research.

3.4.1 Trees

Trees form a natural class of graphs on which to consider the vaccination and firefighter problems because each defended vertex immediately saves its descendants. The low connectivity of trees means they are not as relevant in modeling the interaction of individuals. However, if each vertex represents a larger group that is internally well-connected and has few connections to other groups, then a tree structure is more reasonable. Such examples arise in disease models when considering a household as one vertex. If one individual contracts the disease, then all of the other members of his or her household are very likely to also contract the disease and become infectious. Thus it is reasonable to treat the household as a single unit.

We consider the firefighter problem on trees when the initial outbreak of fire begins at a single root vertex r and when only one vertex can be defended by a firefighter per time step. Note, however, that all of the results extend in a natural way to defending f vertices per time step.

Recall that the vertices in a tree at distance i from the root r are said to be on level i.

Lemma 3.20 (MacGillivray and Wang, Hartnell and Li). *If $a_1, a_2, \ldots$ is an optimal firefighter sequence, then a_i is on level i.*

Proof. Since at time i all vertices higher than level i have either been burned or saved, a_i is on level i or lower. Suppose that i is the least index where a_i is not on level i. Then no vertices are defended on level i, and defending a_i's parent instead of a_i results in a firefighter sequence that saves at least one more vertex than sequence $a_1, a_2, \ldots$. This contradicts the optimality of $a_1, a_2, \ldots$, and so a_i is on level i. □

When a vertex v is defended, all of its descendants are immediately saved. Let $\mathrm{wt}(v)$ denote the number of vertices that are saved (including v) when v is defended. We shall present two different methods for approximating the maximum number of vertices that can be saved in a given tree.

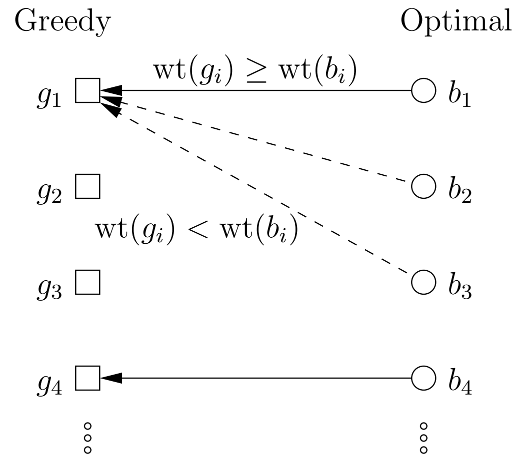

Figure 3.10: Pictorially "charging" b_i to g_j.

3.4.1.1 Greedy Algorithm

A natural method for generating a firefighter sequence is the greedy algorithm: a_i is chosen to be a vertex on level i that has not been saved and that has maximum weight. As shown in Theorem 3.22, the greedy algorithm does not always produce an optimal firefighter sequence. However, we are able to provide some guarantee on the greedy algorithm's performance. The proof given here is essentially the same as that of Hartnell and Li in [10] but with different presentation.

Theorem 3.21 (Hartnell and Li). *On trees, the greedy algorithm generates a firefighter sequence that saves at least half as many vertices as an optimal firefighter sequence.*

Proof. [2] Fix an optimal firefighter sequence $b_1, b_2, \ldots, b_k$ that saves the largest number of vertices, and let $g_1, g_2, \ldots, g_\ell$ be the vertices selected by the greedy algorithm, where b_i and g_i are the vertices defended on level i in the respective sequences. Our approach will be to "charge" each vertex b_i defended in the optimal sequence to a vertex defended by the greedy algorithm. To visualize the concept, we construct a bipartite graph as in Figure 3.10 with the vertices $b_1, b_2, \ldots, b_k$ on the right side and the vertices $g_1, g_2, \ldots, g_\ell$ on the left side. An outgoing arc from b_i to g_j indicates that b_i is being charged to g_j, which we denote by

[2] This version of the proof is based on an idea of Mike Saks.

$b_i \to g_j$. To determine the chargings, compare $\text{wt}(b_i)$ to $\text{wt}(g_i)$. If $\text{wt}(b_i) \leq \text{wt}(g_i)$, then the greedy algorithm is doing well compared to the optimal, and we charge b_i to g_i. If $\text{wt}(b_i) > \text{wt}(g_i)$, then b_i must already be saved, or else the greedy algorithm would pick b_i since it has higher weight. Let g_j be the ancestor of b_i defended by the greedy algorithm, and charge b_i to g_j.

Now we relate the total weight $\mathcal{O}$ of vertices saved by this optimal sequence to the total weight $\mathcal{G}$ of the greedy algorithm by using the standard combinatorial technique of counting in two different ways:

$$\begin{aligned}\mathcal{O} &= \sum_{i=1}^{k} \text{wt}(b_i) \\ &= \sum_{j=1}^{\ell} \left(\sum_{i:b_i \to g_j} \text{wt}(b_i) \right).\end{aligned}$$

To bound $\sum_{i:b_i \to g_j} \text{wt}(b_i)$, note that for $i \neq j$, b_i is a descendant of g_j, and the total weight of all vertices defended under g_j is at most the weight of g_j (the most number of vertices who can be saved below g_j is the number of vertices saved by defended g_j). Thus,

$$\begin{aligned}\sum_{i:b_i \to g_j} \text{wt}(b_i) &\leq \text{wt}(b_j) + \sum_{i \neq j:b_i \to g_j} \text{wt}(b_i) \\ &\leq \text{wt}(g_j) + \text{wt}(g_j) \\ &= 2\text{wt}(g_j),\end{aligned}$$

where we use the observation that in an optimal firefighter sequence, no b_i is a descendant of any other b_m. Thus,

$$\begin{aligned}\mathcal{O} &= \sum_{j=1}^{\ell} \left(\sum_{i:b_i \to g_j} \text{wt}(b_i) \right) \\ &\leq \sum_{j=1}^{\ell} 2\text{wt}(g_j) \\ &= 2\mathcal{G}.\end{aligned}$$

Hence, the greedy algorithm saves at least half as many vertices as an optimal firefighter sequence. □

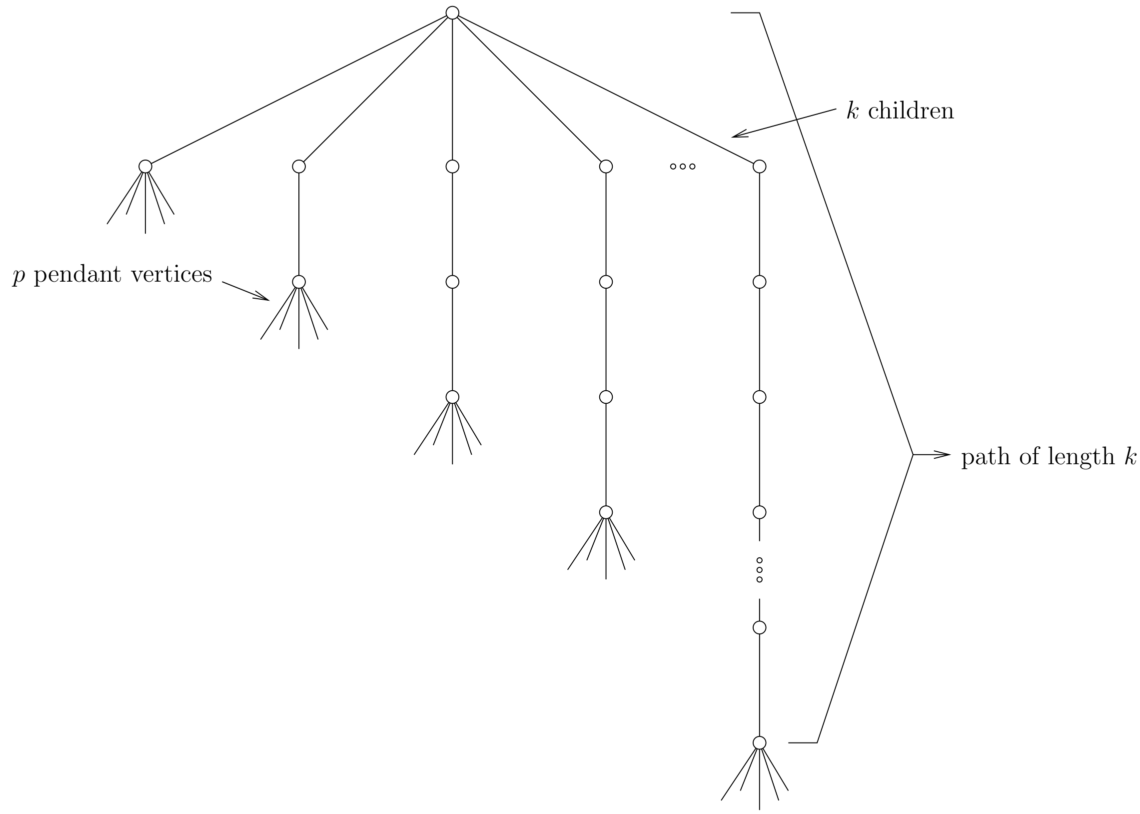

Figure 3.11: Construction showing that asymptotically the greedy algorithm saves 1/2 of the number of vertices saved by an optimal firefighter sequence.

Our proof of the following theorem is different than Hartnell and Li's, and provides some extra insight into the difficulties of strengthening the greedy algorithm.

Theorem 3.22 (Hartnell and Li). *Theorem 3.21 is tight,* i.e., *there are graphs such that the proportion of vertices the greedy algorithm saves compared to an optimal firefighter sequence is arbitrarily close to* 1/2*.*

Proof. The graph shown in Figure 3.11 has $n = 1 + \frac{1}{2}k(k+1) + kp$ vertices. The greedy algorithm (which always defends the rightmost vertex) saves

$$\begin{cases} \frac{k}{2}\left(\frac{k}{2}+1\right) + \frac{k}{2}p + 1, & \text{if } k \text{ is even,} \\ \left\lfloor \frac{k}{2} \right\rfloor \left(\left\lfloor \frac{k}{2} \right\rfloor + 1\right) + \left\lceil \frac{k}{2} \right\rceil + \left\lceil \frac{k}{2} \right\rceil p, & \text{if } k \text{ is odd,} \end{cases}$$

vertices, whereas in the optimal firefighter sequence (which always defends the leftmost vertex), $kp+k$ vertices are saved. Thus, assuming $p \gg k$ and taking p large, the proportion of vertices the greedy algorithm saves is arbitrarily close to 1/2 of the vertices that an optimal firefighter sequence saves. □

It is tempting to try to improve the greedy algorithm by increasing its power while retaining polynomial time. For instance, we could choose a_1 by finding the sequence $a_1, a_2, \ldots, a_k$ that maximizes the weight of the first k vertices in the sequence. Or we could use the greedy algorithm as an approximation for trees of small height in a recursive algorithm: if the height of T is within k of the height of the original tree, then recursively calculate a vaccination sequence; otherwise, use the greedy algorithm as an approximation. Unfortunately, the same set of examples described in Theorem 3.21 show that asymptotically none of these methods save more than 1/2 of the vertices saved by an optimal firefighter sequence. An open question is to find an approximation algorithm which guarantees saving a greater fraction of the optimal number of vertices than 1/2.

3.4.1.2 Linear Programming Approximations for the Firefighter Problem on Trees

MacGillivray and Wang [11] presented an integer program for finding an optimal firefighter sequence $a_1, a_2, \ldots$ for a tree. To each vertex v we associate a boolean variable $x(v)$ that indicates whether v is defended by a firefighter, and we wish to maximize the total number of vertices saved. Let the weight $\mathrm{wt}(v)$ of v denote the number of vertices saved by a firefighter defending v. Thus, $\mathrm{wt}(v)$ is equal to the number of descendants of v plus 1. To ensure that no double-counting occurs in the objective function, we require that no vertex be defended that is already saved. We enforce this requirement by adding the constraint that the sum of $x(v)$ for all ancestors v of a given vertex u and including u is at most 1. It is sufficient to add this constraint only for leaf vertices, since if u is a leaf, then the constraint for all ancestors of u is implied by the constraint for u. Lemma 3.20 gives the constraint that the sum of $x(v)$ for all of the v on a given level is at most 1. We thus have the following integer program of MacGillivray and Wang. Here, we write $v \succ u$ or $u \prec v$ if

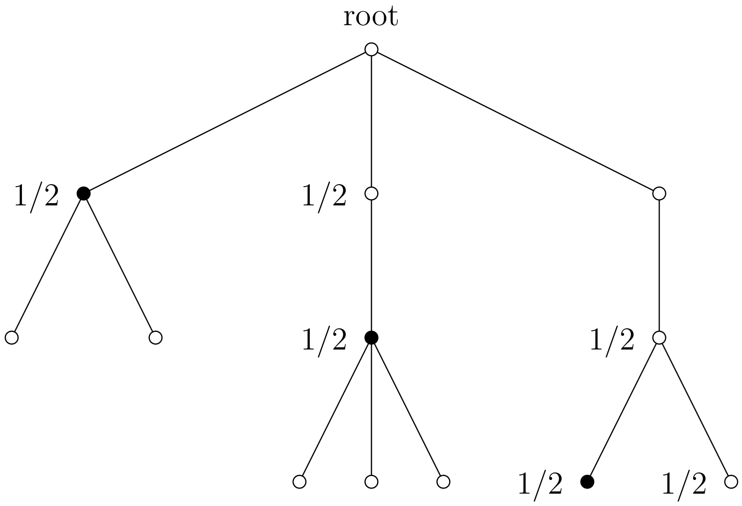

Figure 3.12: In this example on 13 vertices, the LP optimal is 8.5, whereas the IP optimal is 8. The nonzero values of $x(v)$ for the LP optimal solution appear next to the vertices, and the optimal firefighter sequence is indicated with black vertices.

v is an ancestor of u, and we write $v \succeq u$ or $u \preceq v$ if v is an ancestor of u or if $v = u$.

$$\text{maximize} \sum_{v \in V(G)} \text{wt}(v)x(v)$$

$$\text{subject to:} \sum_{v \text{ on level } \ell} x(v) \leq 1, \text{ for each level } \ell, \tag{3.22}$$

$$\sum_{v \succeq u} x(v) \leq 1, \text{ for each leaf } u, \tag{3.23}$$

$$x(v) \in \{0,1\}, \text{ for each vertex } v. \tag{3.24}$$

By relaxing condition (3.24) that $x(v)$ is boolean we obtain a linear program. The linear programming (LP) optimal m^* provides an upper bound to the integer programming (IP) optimal m. In general, the linear program does not have an integral optimal, and so the LP optimal is strictly greater than the IP optimal. Figure 3.12 shows an example where this occurs. It is an open question to bound the size of the "integrality gap," the difference between the linear programming and integer programming optimals.

MacGillivray and Wang showed that by adding the non-linear constraints $x(v)x(u) = 0$ for every non-root vertex v and every descendant u of v, then the optimal solution is integral. In general, solving such a non-linear optimization problem is hard. We take a different

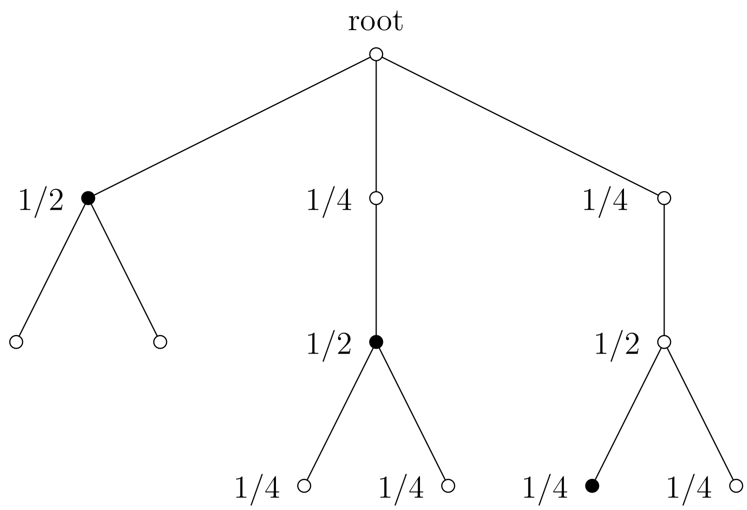

Figure 3.13: In this example on 12 vertices, the LP optimal when using constraint (3.26) is 7.5, whereas the IP optimal is 7. The nonzero values of $x(v)$ for the LP optimal solution appear next to the vertices, and the optimal vaccination sequence is indicated with black vertices.

approach: by adding additional constraints, we will attempt to narrow the integrality gap.

The effect of the leaf constraint (3.23) is that if a vertex v is defended, then none of v's descendants can also be defended. It is tempting to instead use the constraint

$$x(u) + \sum_{v \prec u} x(v) \leq 1, \text{ for each vertex } u. \tag{3.25}$$

However, constraint (3.25) is too restrictive, since it also forbids two descendants on different levels being defended when v is not defended. A weaker approach is to only include in the constraint descendants that are themselves mutually exclusive. All of v's descendants on a given level is one such set. Thus, we add the constraint

$$x(u) + \sum_{\substack{v \preceq u \\ v \text{ on level } i}} x(v) \leq 1, \text{ for each vertex } u \text{ and each level } i \text{ greater than the level of } u. \tag{3.26}$$

Note that with this constraint, we still need the leaf constraint. When using constraint (3.26) on the tree shown in Figure 3.12, the LP optimal is the same as the IP optimal. However, Figure 3.13 shows an example where there is still an integrality gap using constraint (3.26). The tree shown in Figure 3.13 does suggest adding u's ancestors into the summation as

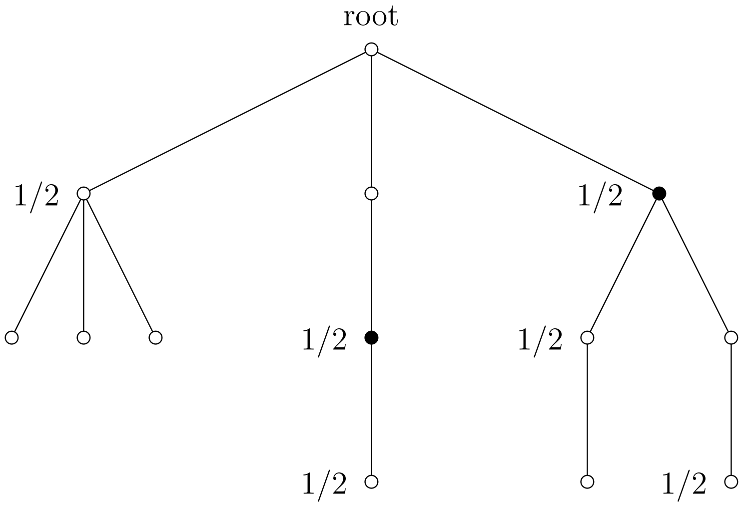

Figure 3.14: In this example on 13 vertices, the LP optimal when using constraint (3.27) is 7.5, whereas the IP optimal is 7. The nonzero values of $x(v)$ for the LP optimal solution appear next to the vertices, and the optimal vaccination sequence is indicated with black vertices.

well. Thus, we have the constraint

$$\sum_{v \succeq u} x(v) + \sum_{\substack{v \preceq u \\ v \text{ on level } i}} x(v) \leq 1, \text{ for each vertex } u \text{ and each level } i \text{ below } u. \tag{3.27}$$

When using constraint (3.27) on the tree in Figure 3.13, the LP optimal is the same as the IP optimal. However, Figure 3.14 shows an example where there is still an integrality gap using constraint (3.27).

For small trees, the LP optimal when using constraint (3.27) is the IP optimal. In fact, we have verified this by computer for trees with up to 11 vertices. We are thus led to

Conjecture 3.23. *The tree in Figure 3.14 is the smallest tree such that the LP optimal when using constraint (3.27) is not the IP optimal.*

For large trees, the LP optimal is very often the IP optimal, and when different is very close. This observation is based on computer experimentation. Approximately 1.68 million trees with 100 vertices were randomly generated, and the LP optimal of MacGillivray and Wang's program, the LP optimal with constraint (3.27), and the IP optimal were calculated. A random tree is generated by starting with the root vertex and adding vertices one at a time, where a vertex is connected to a vertex in the existing tree chosen uniformly at random. Of these trees, 5.22% had the LP optimal of MacGillivray and Wang's program greater than the IP optimal, and the difference was at most 6.34% of the IP optimal. When using constraint (3.27), 0.70% had the LP optimal greater than the IP optimal, and the difference was at most 3.73% of the IP optimal. This data leads us to

Conjecture 3.24. *The ratio of the LP optimal to the IP optimal, with or without constraint (3.27), is bounded for all trees.*

3.4.1.3 Defending One Child Per Burnt Vertex in Trees

One reason that FIREFIGHTER for trees is a difficult problem is because the firefighter response requires a global decision. If we replace the global decision with a local decision, then the problem becomes much easier. In this subsection only, we consider a firefighter response where at each time step we can defend one non-infected, non-defended neighbor of *each* infected vertex. Formally, at time 0, the root r of the tree initially catches fire. Then we may defend one child a_1 of r. The fire then spreads to the non-defended children of the root. We may then defend one child of *each* of those burnt vertices. Let A_i denote the set of vertices initially defended at time i. We thus have a firefighter sequence $A_1, A_2, \ldots, A_h$ of sets of vertices. The sequence contains a set for every level i from 1 to the height h of the tree, but some sets may be empty. Since vertices that are defended must be adjacent to infected vertices, we immediately have the following lemma.

Lemma 3.25. *If a firefighter sequence $A_1, A_2, \ldots, A_h$ is optimal, then the vertices in A_i are on level i.*

As in the global firefighter response, a natural method of generating a firefighter sequence is the greedy algorithm: if u is a burnt vertex on level $i-1$, then we include into A_i a child v of u that has maximum weight.

Theorem 3.26. *On trees, the greedy algorithm generates a firefighter sequence that saves at least half as many vertices as saved by an optimal firefighter sequence.*

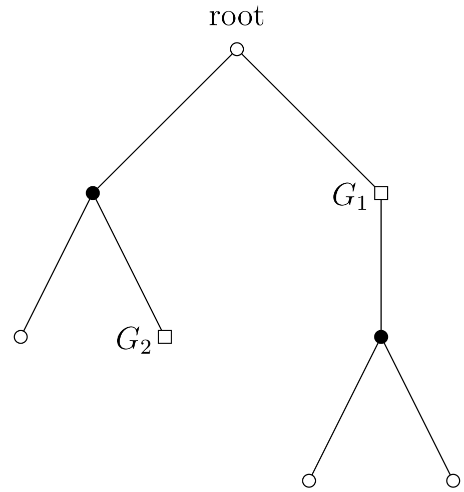

Figure 3.15: When defending one child per burnt vertex in this tree, the greedy algorithm is not optimal. The greedy algorithm defends the square vertices G_1 and G_2, saving 5 vertices, while the optimal firefighter sequence marked with black dots saves 6 vertices.

Proof. We use the same charging technique used in the proof of Theorem 3.21. Fix an optimal firefighter sequence $B_1, B_2, \ldots, B_k$ that saves the largest number of vertices, and let $G_1, G_2, \ldots, G_\ell$ be the vertices selected by the greedy algorithm, where B_i and G_i are the sets of vertices defended on level i in the respective sequences. Let b be a vertex in B_i. If there is an ancestor g of b that is defended by the greedy algorithm, then we charge b to g. If not, then the greedy algorithm defends a child g' of b's parent p that has maximum weight among the children of p. Specifically, $\mathrm{wt}(b) \leq \mathrm{wt}(g')$. In this case, we charge b to g'. The rest of the proof is the same as the proof of Theorem 3.21. □

It is unknown whether Theorem 3.26 is tight; most likely it is not. Figure 3.15 shows an example where the greedy algorithm is not optimal. It is an open problem to determine a constant ρ where the greedy algorithm saves at least a fraction ρ of the number of vertices saved by an optimal firefighter sequence and where this bound ρ is tight.

We can also formulate the problem of finding an optimal firefighter sequence when defending one child per burnt vertex as an integer program. The integer program is very similar to the program for the global firefighter response: to each vertex v we associate a boolean variable $x(v)$ that indicates whether v is in the firefighter sequence, and we wish

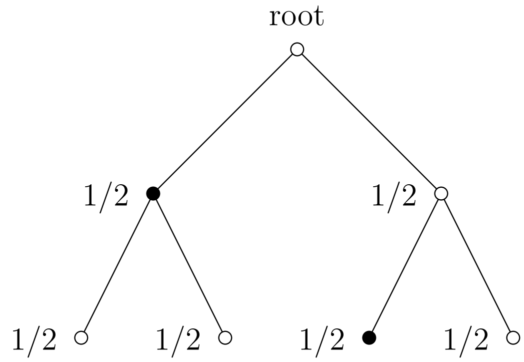

Figure 3.16: This example demonstrates the model of one defended child per burnt vertex. In this example on 7 vertices, the LP optimal is 5, whereas the IP optimal is 4. The nonzero values of $x(v)$ for the LP optimal solution appear next to the vertices, and the optimal firefighter sequence is indicated with black vertices.

to maximize the total number of vertices saved. We also need the leaf constraint (3.23) to ensure that no double-counting occurs in the objective function. We also have the constraint that at most one child per parent may be defended. We thus have the following integer program:

$$\text{maximize} \sum_{v \in V(G)} \text{wt}(v) x(v)$$

$$\text{subject to:} \sum_{v \text{ a child of } u} x(v) \leq 1, \text{ for each vertex } u, \tag{3.28}$$

$$\sum_{v \succeq u} x(v) \leq 1, \text{ for each leaf } u, \tag{3.29}$$

$$x(v) \in \{0, 1\}, \text{ for each vertex } v. \tag{3.30}$$

In general, the LP relaxation is not the IP optimal. Figure 3.16 shows an example where this is the case.

We can also use dynamic programming to find an optimal firefighter sequence in the case of trees when defending one child per burnt vertex.

Dynamic Programming Algorithm When Defending One Child Per Burnt Vertex in Trees

Input: A rooted tree T of height h with root r.

Output: A firefighter sequence $A_1, A_2, \ldots, A_h$.

Recursion Step: If T is of height 1, then set $A_1 := \{v\}$, for an arbitrarily chosen child v of the root.

If $h > 1$, then for each child u of the root, let T_u be the subtree rooted at u. For each subtree, recursively calculate a firefighter strategy $A_2^u, A_3^u, \ldots, A_h^u$. Choose the child v of the root such that the number of vertices saved by the firefighter sequence

$$A_1 = \{v\}, \quad A_2 = \bigcup_{\substack{u \text{ a child of the root} \\ u \neq v}} A_2^u, \quad \ldots, \quad A_h = \bigcup_{\substack{u \text{ a child of the root} \\ u \neq v}} A_h^u$$

is maximum.

This dynamic programming algorithm can be implemented in polynomial time by saving and reusing the optimal firefighter sequence for the subtree rooted at each vertex x. In fact, the algorithm can be run as a one-pass "bottom-up" algorithm, where A_i is calculated beginning at $i = h$ and decrementing i. The same firefighter sequence is optimal for T_x independent of the firefighter placements made higher in the tree, assuming that x is not saved. This property is not true for the global firefighter response, which is why the dynamic programming algorithm in that case requires exponential time.

Theorem 3.27. *The firefighter sequence produced by the dynamic programming algorithm is optimal.*

Proof. We proceed by induction on the height h of the tree T. For $h = 1$, the result is straightforward. So suppose that $h > 1$. Let $B_1 = \{w\}, B_2, \ldots, B_h$ be an optimal firefighter sequence, and let $A_1 = \{v\}, A_2, \ldots, A_h$ be the firefighter sequence produced by the dynamic programming algorithm. Consider each subtree T_u rooted at a child u of the root. Define $B_i^u = B_i \cap V(T_u)$ for $u \neq w$ and $2 \leq i \leq h$, and let A_i^v $(2 \leq i \leq h)$ be the firefighter sequence recursively calculated by the dynamic programming algorithm for the subtree T_u.

Notice that $A_i^u = A_i \cap V(T_u)$ for $u \neq v$. The height of T_u is at most $h-1$, and by induction the firefighter sequence $A_2^u, A_3^u, \ldots, A_h^u$ is optimal. Thus, for any $u \neq w$, the sequence $A_2^u, A_3^u, \ldots, A_h^u$ saves as many vertices in T_u as $B_2^u, B_3^u, \ldots, B_h^u$.

Form the firefighter sequence $A_1', A_2', \ldots, A_h'$ by choosing w for A_1 instead of v. Thus,

$$A_1' = \{w\}, \quad A_2' = \bigcup_{\substack{u \text{ a child of the root} \\ u \neq w}} A_2^u, \quad \ldots, \quad A_h' = \bigcup_{\substack{u \text{ a child of the root} \\ u \neq w}} A_h^u.$$

By the way v was chosen, the sequence $A_1, A_2, \ldots, A_h$ saves at least as many vertices as $A_1', A_2', \ldots, A_h'$. But the number of vertices saved by $A_1', A_2', \ldots, A_h'$ is

$$\mathrm{wt}(w) + \sum_{\substack{u \text{ a child of the root} \\ u \neq w}} (\# \text{ of vertices saved in } T_u \text{ by } A_2^u, A_3^u, \ldots, A_h^u),$$

which is at least as many vertices as $B_1, B_2, \ldots, B_h$ saves. Thus, the sequence $A_1, A_2, \ldots, A_h$ saves at least as many vertices as $B_1, B_2, \ldots, B_h$, and so $A_1, A_2, \ldots, A_h$ is an optimal firefighter sequence for T. □

3.4.2 Other Questions

There are many avenues for future work in models of responses to disease and fire spread. For infinite graphs, we can ask the same question as for the infinite square grids: What is the minimum number of firefighters needed per time step so that only a finite number of vertices are burned? Percolation is a related topic whose methods may also apply here. For trees, it would be interesting to have an exact characterization of when the greedy algorithm is optimal and when it is not. Of course, bounding the size of the integrality gap as stated in Conjecture 3.24 is an open question. Finding more restrictive constraints that additionally reduce the integrality gap would also be interesting.

From the viewpoint of a bioterrorist or arsonist, one would like to find the most vulnerable vertices in a graph G. A vertex v is most vulnerable if a disease outbreak starting at v infects the most vertices G given an optimal vaccination response. Can the most vulnerable vertices in a graph be determined without knowing the optimal vaccination response? Perhaps they could then be preemptively vaccinated. From the viewpoint of a network architect, we would like to design graphs that are resistant to such attacks. Similar questions can also be asked if there are k initial outbreaks of disease.

The inclusion of weights on vertices is a natural generalization. For instance, some people such as health care workers might be more important to protect since they are necessary to implement the vaccination strategy. In the firefighter model, areas with homes might be more important than unpopulated areas. The inclusion of speeds on edges is another natural generalization, since the rate of transmission of a disease might differ between pairs of individuals or since the rate of fire spread might vary between two regions, for instance, because of density of underbrush. The question with these extensions is still to determine a vaccination or firefighter response that saves the most vertices, but here "most" means according to the weights on vertices.

Finally, MacGillivray and Wang [11] observed that the firefighter problem can be viewed as a one-player game. Suppose that the fire has a choice, too: the fire can only spread to d neighbors each time step. This forms a two-player game. What strategy should the firefighters use to minimize the number of burned vertices?

References

[1] R. Durrett and S. A. Levin, Can Stable Social Groups Be Maintained By Preferential Imitation Alone?, *Journal of Economic Behavior and Organization*, to appear.

[2] J. M. Epstein, D. A. T. Cummings, S. Chakravarty, R. M. Singa, and D. S. Burke, Toward a Containment Strategy for Smallpox Bioterror: An Individual-Based Computational Approach, The Brookings Institute Center on Social and Economic Dynamics Working Paper No. 31, December 2002.

[3] S. Eubank, H. Guclu, V. S. A. Kumar, M. V. Marathe, A. Srinivasan, Z. Toroczkai, N. Wang, and the EpiSims Team, `http://episims.lanl.gov`, April 22, 2004.

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

[5] S. Finbow, A. King, G. MacGillivray, and R. Rizzi, The Firefighter Problem for Graphs of Maximum Degree Three, manuscript, 2004.

[6] P. Fogarty, *Catching the Fire on Grids*, M.Sc. Thesis, Department of Mathematics, University of Vermont, 2003.

[7] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman and Company, New York, 1979.

[8] GNU Linear Programming Kit, `http://www.gnu.org/software/glpk/glpk.html`.

[9] B. Hartnell, Firefighter! An Application of Domination, presentation, Twentieth Conference on Numerical Mathematics and Computing, University of Manitoba in Winnipeg, Canada, Sept. 1995.

[10] B. Hartnell and Q. Li, Firefighting on Trees: How Bad is the Greedy Algorithm?, *Congressus Numerantium*, 145 (2000), 187-192.

[11] G. MacGillivray and P. Wang, On the Firefighter Problem, *J. Combin. Math. Combin. Comput.*, 47 (2003), 83-96.

[12] J. M. Read and M. J. Keeling, Disease Evolution on Networks: The Role of Contact Structure, *Proc. Roy. Soc. Lond. B*, 270 (2003), 699-708.

[13] P. Wang and S. A. Moeller, Fire Control on Graphs, *J. Combin. Math. Combin. Comput.*, 41 (2002), 19-34.