GRAPH-THEORETIC MODELS OF SPREAD AND COMPETITION

BY STEPHEN G. HARTKE

A dissertation submitted to the

Graduate School—New Brunswick

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

Graduate Program in Mathematics

Written under the direction of

Fred S. Roberts

and approved by

_____

_____

_____

_____

New Brunswick, New Jersey

October, 2004

ABSTRACT OF THE DISSERTATION

Graph-theoretic Models of Spread and Competition

by Stephen G. Hartke

Dissertation Director: Fred S. Roberts

We consider three graph-theoretical models of spread and competition motivated by biological applications and also by spread of opinions in social networks and fires in geographical areas. We modify the voter model of Clifford, Sudbury, Holley, and Liggett by introducing confidence levels. In the voter model, a group of voters has opinions "yes" or "no," interpreted as infected or non-infected in a disease application. As time progresses each voter's opinion is influenced by his or her neighbors, with the confidence level of a voter determining how quickly the voter reconsiders his or her opinion (how quickly a person might change disease state). We show that the voter model with confidence levels always results in a uniform opinion, and we determine the probability of each uniform opinion based on the initial opinions and–what is unusual in this subject–on the structure of the underlying graph.

We also consider a perfectly contagious disease, where vertices adjacent to infected vertices become infected at every discrete time step. The only intervention allowed is a limited number of vaccinations per time step. This model of disease spread is equivalent to a model of fire spread introduced by Hartnell where firefighters correspond to vaccinations. We prove a conjecture of Wang and Moeller about the number of firefighters needed per time step to contain an outbreak starting at a single vertex in an infinite square grid with dimension at least three. We then show that no constant number of firefighters per

time step is sufficient to contain every finite outbreak of fire. We also present a new proof of MacGillivray and Wang's result that finding an optimal firefighter strategy is NP-complete for general graphs.

Finally, we study questions arising from competition between species and phylogenetic tree reconstruction, considering elimination procedures for the competition number and the phylogeny number of a graph. We provide a simpler proof of Kim and Roberts' theorem that their elimination procedure calculates the competition number for "kite-free" graphs. We answer a question in the literature by showing that their procedure does not calculate the competition number for all graphs. We introduce an elimination procedure for the phylogeny number and show it computes the phylogeny number for "kite-free" graphs, but that it does not calculate the phylogeny number for all graphs.

Acknowledgements

I first thank my advisor Fred Roberts for providing guidance, encouragement, and inspiration while working on this thesis. His insight and patience while listening and reading many revisions enabled me produce this thesis. I also thank József Beck, Doron Zeilberger, and Yehuda Vardi for serving on my thesis committee.

I thank all of the faculty at Rutgers from whom I have taken classes or have had discussions during my years at Rutgers, particularly József Beck, Jeff Kahn, Doron Zeilberger, and Endre Boros. I also thank my fellow graduate students for lively discussions and encouragement, particularly Eva Curry, Pieter Blue, Nick Weininger, Klay Kruczek, and Eric Sundberg. I thank Mike Develin for countless evenings watching the fog and discussing life and mathematics.

Several people have provided specific suggestions for improving my thesis. I thank Yehuda Vardi for suggesting the waiting function in the voter model with confidence levels, Michael Weingart for pointing out the simple proof of Lemma 2.6, Larry Shepp for helpful discussions about semi-Markov processes with infinite holding times, Mike Develin for joint work on the firefighter problem on grids with dimension at least three, and Suh-Ryung Kim for explaining the strength of the kite-free property in graphs.

I gratefully acknowledge being supported from many sources while I was a graduate student. I was a National Science Foundation Graduate Research Fellows and a National Defense Science and Engineering Graduate Fellow, I received a DIMACS Graduate Student Research Award, and I received support from the National Science Foundation through DIMACS under grants EIA-0205116, DBI-9982983, and SBR-9709134. DIMACS has also funded the Graduate Student Combinatorics Seminar, which has been a very successful catalyst for discussion among the graduate students.

Finally, I thank for my parents for their constant love and support.

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Overview

Discrete mathematics has many applications to other fields. In this thesis, we are going to consider several models that are motivated by biological applications, mainly modeling the spread of diseases and interventions attempting to contain this spread, and modeling competition between species. The models that we investigate also arise in other areas, such as the spread of opinions in social networks and the spread of fire in geographic areas, and the results that we present are applicable to an array of topics.

This thesis consists of three major chapters. In Chapter 2 we present a modification of the voter model by introducing the concept of confidence levels. In the voter model, we have a group of voters and each voter has an opinion about some issue, either 1 ("yes") or 0 ("no"). By considering the voters as vertices and the neighbor relation as edges, we naturally have an underlying graph G. We define a stochastic process on G where as time progresses each voter's opinion is influenced by his or her neighbors. In a disease interpretation, 1 signifies that an individual is infected with a disease and 0 indicates that the individual is uninfected. The voter model was introduced independently by Clifford and Sudbury [2] and by Holley and Liggett [9]. The basic conclusion is that the model always results in a uniform opinion and that the probability of ending with a given opinion is essentially independent of the graph structure. Motivated by ideas of Hoffman [8] and Roberts [13], we introduce a notion of confidence into the voter model. The confidence that a voter has in his or her current opinion determines how quickly the voter reconsiders his or her opinion. Confidence levels can also be interpreted as resistance to infection by a disease. We show that the voter model with confidence levels always results in a uniform

opinion, and we determine the probability of each outcome (uniform 1 or 0) based on the initial opinions and–what is different than the classic voter model–on the structure of the graph.

In Chapter 3 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. We consider a simple spread mechanism on a graph G: 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 per time step. The main question we investigate is finding an optimal strategy for vaccinations in order to minimize the total number of infected vertices. We are primarily interested in the situation when there is only one initially infected root vertex, and there are exactly f vaccinations allowed per time step. The model of disease spread that we consider in Chapter 3 is equivalent to a model of fire spread introduced by Hartnell [7]. In this model, fire spreads from an outbreak to adjacent vertices at each time step. In response, firefighters can be deployed to defend vertices and prevent the fire from spreading to them. The motivating question is again to find an optimal sequence of defended vertices that minimizes the total number of burnt vertices. In light of the literature on the firefighter model, we will primarily use the terminology of that model in presenting our results. In the case of infinite square grids, we prove a conjecture of Wang and Moeller [17] about the number of firefighters needed per time step to contain an outbreak starting at a single vertex. Motivated by work of Fogarty [4] on two-dimensional square grids, we also show that no constant number of firefighters per time step is sufficient to contain every finite outbreak of fire. MacGillivray and Wang [11] proved that the problem of finding an optimal sequence of vertices defended by firefighters is NP-complete for general graphs. We present a new proof of this result which holds under the assumption of small average vertex degree, which is more realistic in the disease applications than the larger average vertex degree of the MacGillivray-Wang proof. Finally, we discuss an approximation technique for the problem on trees.

In Chapter 4 we study elimination procedures for the competition number and the phylogeny number of a graph. Given an acyclic digraph D, the competition graph $C(D)$ is defined to be the undirected graph with $V(D)$ as its vertex set and where vertices x and y are adjacent if there exists another vertex z such that the arcs (x, z) and (y, z) are both present in D. The competition number $k(G)$ for an undirected graph G is the least number r such that there exists an acyclic digraph F on $|V(G)|+r$ vertices where $C(F)$ is G along with r isolated vertices. Kim and Roberts [10] introduced an elimination procedure for the competition number and asked whether the procedure calculated the competition number for all graphs. We provide a simpler proof of Kim and Roberts' theorem that their elimination procedure calculates the competition number for the so-called "kite-free" graphs. However, we answer their question for all graphs in the negative by demonstrating a graph where the elimination procedure does not calculate the competition number.

We also study in Chapter 4 a variant of the competition number known as the phylogeny number. Given an acyclic digraph D, the phylogeny graph $P(D)$ is defined to be the undirected graph with $V(D)$ as its vertex set and with adjacencies as follows: two vertices x and y are adjacent if one of the arcs (x, y) or (y, x) is present in D, or if there exists another vertex z such that the arcs (x, z) and (y, z) are both present in D. Phylogeny graphs were introduced by Roberts and Sheng [14] from an idealized model for reconstructing phylogenetic trees in molecular biology and are closely related to competition graphs. The phylogeny number $p(G)$ for an undirected graph G is the least number r such that there exists an acyclic digraph D on $|V(G)| + r$ vertices where G is an induced subgraph of $P(D)$. We introduce an elimination procedure for the phylogeny number analogous to the elimination procedure of Kim and Roberts [10] for the competition number and show that our elimination procedure computes the phylogeny number for "kite-free" graphs. We also show that the elimination procedure does not calculate the phylogeny number for all graphs.

1.2 Basic Definitions

We assume that basic concepts in graph theory, combinatorics, probability, discrete optimization, and complexity theory are understood by the reader. We will present basic terminology and notation here, and introduce other terms and notation throughout the thesis as necessary.

1.2.1 Graph Theory

Diestel [3] and West [18] provide good introductions to graph theory, and Van Lint and Wilson [16] is a useful reference for general combinatorics. A *graph* $G = (V, E)$ consists of a set V of *vertices* and a set E of unordered pairs of vertices called *edges*. We often write $V(G)$ and $E(G)$ to denote the vertex and edge set, respectively, to emphasize the graph G. If $e = (v, w)$ is an edge of G, then we call v and w the *endvertices* or *endpoints* of e. We also say that v is *adjacent* to w and often write $v \sim w$ or that vw (and wv) is an edge in G. All of the vertices that are adjacent to v are called *neighbors* of v and the set of neighbors is denoted $N(v)$ or $N_G(v)$ to emphasize the graph G. If B is a subset of the vertices of G, then $N(B) = \cup_{v \in B} N(v)$. Sometimes $N(v)$ and $N(B)$ are called the *open neighborhood* of v and B, respectively. The *closed neighborhood* $N[v]$ of v is $N(v) \cup \{v\}$ and the closed neighborhood $N[B]$ of B is $N(B) \cup B$. The *degree* $\deg(v)$ (or $\deg_G(v)$) of a vertex v is $|N(v)|$. If $\deg(v) = \deg(w)$ for all vertices $v, w \in V(G)$, then G is *regular*. A vertex v with $\deg(v) = 0$ is called an *isolated vertex*, and a vertex w with $\deg(w) = 1$ is called a *pendant*. A *multigraph* is a graph with multiple edges between two vertices. A *loop* is an edge where the two endvertices are the same. Most of the graphs that we consider are *simple graphs* that do not have multiple edges or loops.

A graph $H = (V', E')$ is a *subgraph* of $G = (V, E)$ if $V \subseteq V'$ and $E \subseteq E'$. H is an *induced subgraph* of G if for all vertices $v, w \in V'$, v is adjacent to w in H if and only if v is adjacent to w in G. If H is an induced subgraph, then we say that H is the subgraph of G *induced by* V'. H is a *spanning subgraph* of G if $|V'| = |V|$. For convenience, we will sometimes describe a subgraph H of a graph G only as "consisting of" certain edges of G. It is understood that H has no isolated vertices: the vertices of H are only the endpoints

of edges in H.

A sequence of vertices $v_1, v_2, \ldots, v_k$ is a *path* from v_1 to v_k if all of the v_i are distinct and if v_i is adjacent to v_{i+1} for $1 \leq i < k$. The *length* of a path $v_1, v_2, \ldots, v_k$ is $k-1$. A *cycle* is a path with the additional property that v_k is adjacent to v_1. We call a graph with no cycles *acyclic*. If there is a path from v to w for every pair v, w of vertices in G, then we say that G is *connected*. A subgraph H of G is a *connected component* of G if H is connected and is a maximal subgraph with this property. The *distance* $d(v, w)$ from v to w is the length of the shortest path from v to w if v and w are in the same connected component, and ∞ otherwise.

A *rooted graph* (G, r) is a graph G where we have distinguished a specific vertex r called the *root*. The vertex set $V(G)$ can be partitioned as $V(G) = V_\infty \cup V_0 \cup V_1 \cup \cdots$, where $V_0 = \{r\}$, $V_i = \{v \in V : d(v, r) = i\}$ for $i > 0$, and $V_\infty = \{v \in V : v \text{ is not in the same connected component as } r\}$. The vertices in V_i for $i \geq 0$ are said to be *on level* i or *at level* i from the root.

A *tree* is an acyclic connected graph. We call the pendants in a tree *leaves*. Let (T, r) be a rooted tree, and for a nonroot vertex v, let $v = v_1, v_2, \ldots, v_k = r$ be a shortest path from v to the root r. We say that v_2 is the *parent* of v, v is a *child* of v_2, v_i is an *ancestor* of v for $2 \leq i \leq k$, and that v is a *descendant* of v_i for $2 \leq i \leq k$.

A *complete graph* is a graph where every vertex is adjacent to every other vertex. A clique H of G is a subgraph of G that is also complete. An *edge clique covering* of G is a collection $\{H_1, H_2, \ldots, H_t\}$ of cliques of G such that every edge of G appears in at least one clique H_i.

The *disjoint union* of two graphs $G = (V, E)$ and $H = (V', E')$ where $V \cap V' = \varnothing$ and $E \cap E' = \varnothing$ is the graph $G \cup H = (V'', E'')$, where $V'' = V \cup V'$ and $E'' = E \cup E'$. We let I_r denote the graph with r isolated vertices. Thus, $G \cup I_r$ is the graph formed by adding r isolated vertices to G.

A *directed graph*, or *digraph*, $D = (V, A)$ consists of a set V of vertices and a set A of ordered pairs of vertices called *arcs* or *directed edges*. We often write $V(D)$ and $A(D)$ to denote the vertex and arc set, respectively, to emphasize the digraph D. If $e = (v, w)$

is an arc of D, then we say that e is *directed* or *oriented* from v to w, and we write vw is an arc in G. The *out-neighborhood* $N_D^{\text{out}}(v)$ of a vertex v is $\{w \in V(D) : vw \in A(D)\}$, and the *in-neighborhood* $N_D^{\text{in}}(v)$ is $\{w \in V(D) : wv \in A(D)\}$. These sets are sometimes called the *open out-neighborhood* and the *open in-neighborhood*, respectively. The *closed out-neighborhood* $N_D^{\text{out}}[v]$ is $N^{\text{out}}(v) \cup \{v\}$, and the *closed in-neighborhood* $N_D^{\text{in}}[v]$ is $N^{\text{in}}(v) \cup \{v\}$. If B is a subset of the vertices of D, then the corresponding definitions are $N_D^{\text{out}}(B) = \cup_{v \in B} N_D^{\text{out}}(v)$, $N_D^{\text{in}}(B) = \cup_{v \in B} N_D^{\text{in}}(v)$, $N_D^{\text{out}}[B] = \cup_{v \in B} N_D^{\text{out}}(v) \cup B$, and $N_D^{\text{in}}[B] = \cup_{v \in B} N_D^{\text{in}}(v) \cup B$. If the digraph D is clear, then we will often drop the subscripted "D" from the neighborhood notation. The *in-degree* $\deg_D^{\text{in}}(v)$ of a vertex v is $\left|N_D^{\text{in}}(v)\right|$, and the *out-degree* $\deg_D^{\text{out}}(v)$ is $\left|N_D^{\text{out}}(v)\right|$. A *sink* is a vertex with out-degree zero, and a *source* is a vertex with in-degree zero.

A sequence of vertices $v_1, v_2, \ldots, v_k$ is a *directed path* from v_1 to v_k if all of the v_i are distinct and if $v_i v_{i+1}$ is an arc in D for $1 \leq i < k$. A *directed cycle* is a directed path with the additional property that $v_k v_1$ is an arc in D. We call a digraph with no directed cycles *acyclic*. If there is a directed path from v to w for every pair v, w of vertices in D, then we say that D is *strongly connected*.

1.2.2 Markov and Semi-Markov Processes

Roberts [12] and Gallager [5] contain introductions to Markov chains and semi-Markov processes. A *Markov chain* is a discrete-time stochastic process $\{X_n\}_{n \geq 0}$ such that the probability of X_n being in state j is dependent only on the state X_{n-1} and independent of n and $X_0, X_1, \ldots X_{n-2}$; formally,

$$\Pr\{X_n = j \mid X_0, X_1, \ldots, X_{n-1}\} = \Pr\{X_n = j \mid X_{n-1}\} = \Pr\{X_1 = j \mid X_0\}.$$

The probabilities $p_{ij} = \Pr\{X_n = j \mid X_{n-1} = i\}$ are called *transition probabilities*, and the matrix P with entries p_{ij} is called the *transition matrix.* If there are only a finite number of states that X_n can take on, then we say that $\{X_n\}_{n \geq 0}$ is a *finite-state* Markov chain. An *ergodic set* is a nonempty set $\mathcal{E}$ of states such that $\Pr\{X_n = j \mid X_{n-1} = i\} = 0$ for all states $i \in \mathcal{E}$ and $j \notin \mathcal{E}$, and such that no proper subset of $\mathcal{E}$ also has this property. A Markov chain is *ergodic* if the entire set of states is an ergodic state. A *regular* Markov

chain is an ergodic finite-state Markov chain where there exists a positive integer N such that $\Pr\{X_{N+n} = j \mid X_n = i\} > 0$ for all states i, j. If $\{X_n\}_{n\geq 0}$ is a regular Markov chain, then $\lim_{n\to\infty} \Pr\{X_n = j\}$ exists and is independent of the initial condition X_0. The distribution of X_n as n gets large tends to the *stationary distribution* of $\{X_n\}$. An *absorbing state* is an ergodic set of size one. If $\{X_n\}_{n\geq 0}$ is a finite-state Markov chain where the only ergodic sets are absorbing states, then $\{X_n\}_{n\geq 0}$ is *absorbing* and $\lim_{n\to\infty} \Pr\{X_n \text{ is in an absorbing state}\} = 1$.

A *semi-Markov process* $\{Z_t\}_{t\geq 0}$ is a continuous-time stochastic process with an associated discrete process $\{T_n\}_{n\geq 0}$. We use the convention that $T_0 = 0$. The times $\{T_n\}$ indicate when Z_t changes state. For $n > 0$ and $T_{n-1} \leq t < T_n$, Z_t is constant. At time T_n, Z_t transitions into a new state according to the probability

$$\Pr\left\{Z_{T_n} = j \mid Z_{T_{n-1}}, Z_{T_{n-2}}, \ldots, Z_0\right\} = \Pr\left\{Z_{T_n} \mid Z_{T_{n-1}}\right\}.$$

The probabilities $p_{ij} = \Pr\left\{Z_{T_n} = j \mid Z_{T_{n-1}} = i\right\}$ are called *transition probabilities.* The length of the time interval $T_n - T_{n-1}$ is a random variable with distribution determined by the state $Z_{T_{n-1}}$. If $Z_{T_{n-1}} = i$, then this random variable, which we shall call H_i, is known as the *holding time in state i.* The *embedded Markov chain* $\left\{\hat{X}_n\right\}_{n\geq 0}$ is defined as $\hat{X}_n = Z_{T_n}$. The embedded Markov chain $\left\{\hat{X}_n\right\}_{n\geq 0}$ has the same transition probabilities as the semi-Markov process $\{Z_t\}_{t\geq 0}$. If $\{Z_t\}$ has a finite number of states and the expected holding time is finite for each state, then the limiting probability of Z_t being in state j is

$$\lim_{t\to\infty} \Pr\{Z_t = j\} = \frac{\sum_{\text{states } i} \left(\mathrm{E}[H_i] \lim_{n\to\infty} \Pr\left\{\hat{X}_n = j\right\}\right)}{\sum_{\text{states } i} \mathrm{E}[H_i]}.$$

A special case of a semi-Markov process is when the holding times are independent identically distributed exponential random variables. In this case, the process is a *continuous-time Markov process.*

1.2.3 Complexity Theory and NP-completeness

Garey and Johnson [6] is the standard text for the theory of NP-completeness. A *decision problem* L is a "yes" or "no" question about a specific type of input. An *instance* I of

a decision problem is the question asked about a specific input of length n. A decision problem L is said to be in the complexity class P or *solvable in polynomial time* if there exists a deterministic algorithm A such that if the answer to the instance I is "yes," then A produces this answer in a deterministic running time bounded by a polynomial in n. If the answer to the instance is "no," then A either returns "no" or does not halt. A decision problem L is said to be in the complexity class NP if there exists a nondeterministic algorithm A such that if the answer to the instance I is "yes," then A produces this answer in a nondeterministic running time bounded by a polynomial in n. If the answer to the instance is "no," then A either returns "no" or does not halt. A decision problem K is polynomial-time *reducible* to a decision problem L if there exists a transformation f computable in polynomial time that maps an instance of K to an instance of L with the same answer. A decision problem L in NP is NP-*complete* if every decision problem K in NP is reducible to L.

A boolean form φ in *conjunctive normal form* is written as a conjunction of disjunctive clauses:

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

A truth assignment τ of the boolean variables $x_1, x_2, \ldots, x_k$ *satisfies* φ if φ evaluates to true when substituting the truth values into φ for the variables. The boolean formula satisifiability problem SAT is to determine whether a boolean formula φ in conjunctive normal form has a satisfying truth assignment τ for the variables $x_1, x_2, \ldots, x_k$. Cook's Theorem states that SAT is NP-complete.

1.2.4 Linear and Integer Programming

Chvátal [1] and Schrijver [15] discuss the theory and background of linear and integer programming. A *linear program* L is the optimization problem

$$\text{minimize} f(x_1, x_2, \ldots, x_k) \tag{1.1}$$

$$\begin{aligned} \text{subject to} & g_1(x_1, x_2, \ldots, x_k) \asymp_1 c_1 \\ & g_2(x_1, x_2, \ldots, x_k) \asymp_2 c_2 \\ & \vdots \\ & g_\ell(x_1, x_2, \ldots, x_k) \asymp_\ell c_\ell \end{aligned} \tag{1.2}$$

where $x_1, x_2, \ldots, x_k$ are real variables, $f, g_1, g_2, \ldots, g_\ell$ are linear functions of $x_1, x_2, \ldots, x_k$, and $\asymp_i \in \{=, <, >, \leq, \geq\}$. The function f is known as the *objective function,* and a linear program also results by replacing "minimize" in (1.1) with "maximize." The conditions (1.2) are known as the *constraints.* If $x_1, x_2, \ldots, x_k$ satisfy the constraints and minimize the objective function, then $x_1, x_2, \ldots, x_k$ is an *optimal solution* and $f(x_1, x_2, \ldots, x_k)$ is an *optimal value.* Solving a linear program can be done in polynomial time and in practice can be done quickly.

An *integer program* I is a linear program with the added constraints that x_i is an integer for $1 \leq i \leq k$. The *linear program relaxation* L is the linear program formed from I by removing the integrality constraints. If m is the optimal value of I and m^* is the optimal value of L, then $m \leq m^*$ if I is a minimization problem, and $m \geq m^*$ if I is a maximization problem. The difference $|m^* - m|$ is known as the *integrality gap.* In general, solving an integer program is NP-complete.

References

[1] V. Chvátal, *Linear Programming*, W. H. Freeman and Company, New York, 1983.

[2] P. Clifford and A. Sudbury, A Model for Spatial Conflict, *Biometrika*, 60 (1973), 581-588.

[3] R. Diestel, *Graph Theory*, 2nd ed., Graduate Texts in Mathematics, no. 173, Springer-Verlag, New York, 1997.

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

[5] R. G. Gallager, *Discrete Stochastic Processes*, Kluwer Academic Publishers, Boston, 1996, 174-178.

[6] 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.

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

[8] F. Hoffman, presentation at the Twenty-Sixth Southeastern International Conference on Combinatorics, Graph Theory, and Computing. Held at Florida Atlantic University in Boca Raton, Florida, March 6-10, 1995.

[9] R. Holley and T. M. Liggett, Ergodic Theorems for Weakly Interacting Systems and the Voter Model, *Ann. Probab.*, 3 (1975), no. 4, 643-663.

[10] S.-R. Kim and F. S. Roberts, The Elimination Procedure for the Competition Number, *Ars Combinatoria,* 50 (1998), 97-113.

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

[12] F. S. Roberts, *Discrete Mathematical Models, with Applications to Social, Biological, and Environmental Problems*, Prentice-Hall, Upper Saddle River, New Jersey, 1976.

[13] F. S. Roberts, unpublished notes, 1999.

[14] F. S. Roberts and L. Sheng, Phylogeny Numbers, *Discrete Applied Mathematics*, 87 (1998), 213-228.

[15] A. Schrijver, *Theory of Linear and Integer Programming*, John Wiley and Sons, New York, 1986.

[16] J. H. van Lint and R. M. Wilson, *A Course in Combinatorics*, Cambridge University Press, Cambridge, United Kingdom, 1992.

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

[18] D. B. West, *Introduction to Graph Theory*, Prentice Hall, Upper Saddle River, New Jersey, 1996.

Chapter 2

The Voter Model with Confidence Levels

2.1 Introduction

Groups of people often have to decide together on some issue or course of action. We wish to form a model from which we can make conclusions about the group as a whole. In the model, we will incorporate the group dynamics on a local scale, including how and when a person changes his or her opinion and how other people influence this decision. The *voter model* is one stochastic process that models such interactions. In the voter model, there is a group of voters, each of whom has an opinion 1 or 0 representing *yes* or *no* on some issue. Each voter has relationships to a subset of the other voters, whom we call neighbors, and as time passes, each voter's opinion is influenced by its neighbors. We wish to know how the voters' opinions evolve as a whole. Do they ever come to a consensus on the issue? If consensus occurs, which outcome (1 or 0) is more likely, given every voter's initial opinion? If the vertices of a graph G represent voters and the edges neighbor relationships between voters, then how does the structure of G affect the outcome?

Our work on the voter model was also motivated by an interest in the spread of infectious diseases. Here, the opinions correspond to disease states and a disease spreads from individuals to "neighbors," individuals with whom someone has social contact. We wish to determine how the distribution of infected individuals evolves over time. Do we ever end up in the state when no one is infected or the state where everyone is infected? How does the structure of the edge neighbor relationships affect the outcome? The voter model does not precisely capture disease dynamics since being infected and uninfected is not symmetric in the same way the 1 and 0 opinions are. However, insights into the voter model may lead to insights into more complicated epidemiological models.

The voter model was introduced independently by Clifford and Sudbury [3] and by Holley and Liggett [16]. Clifford and Sudbury were interested in modeling spatial conflicts and geographic dominance between two competing species, and Holley and Liggett viewed the voter model as describing the configuration of spins in a particle system. As the name suggests, the voter model also can be viewed as representing the opinions of a group of voters. Our terminology will come from this last interpretation.

In the extensive literature on the voter model, the limiting behavior of the process is completely described. Unfortunately, the dependence of the limiting behavior on the graph G is minimal and hence not very interesting. Because of the interaction mechanism usually used in such models for determining how a voter is influenced by its neighbors, the final outcome is based only a simple graph theoretic property: the sum of the degrees of the vertices where the voters' initial opinions are 1. In this chapter, we present a different interaction mechanism involving a voter's *confidence level*. In this modified voter model, the structure of G has a large impact on evolution of the voters' opinions. When G is finite and connected, consensus always occurs, and we can explicitly calculate how likely each outcome is. The remarkable feature of the voter model with confidence levels is that determining the likelihood of each outcome remains tractable *despite* this dependence on the structure of G.

The standard reference for the voter model and related interacting particle systems is Liggett [20] and its continuation [21]. Griffeath [14] and Durrett [9] are also widely used sources that provide slightly different perspectives. While the previous authors consider the voter model on infinite graphs, particularly the square lattices in various dimensions, Aldous and Fill [1] is a very readable account of the theory on finite graphs. Also on finite graphs, Donnelly and Welsh [7] consider the probability of each outcome and the time needed to reach consensus.

The area of social influence has an extensive literature on the models of opinion formulation. Poljak and Sûra [25] developed a model where each person has one of a finite set of opinions, each person has some measure of influence on the opinion of every other

person, and these influences are symmetric. At each discrete time step, a person's opinion is updated to the opinion held by the most neighbors, weighted by their influences. DeGroot [5] and French [12] proposed a similar model with opinions taken from a real interval, and where a person's opinion is updated to an average of his or her neighbor's opinions, the average being weighted according to the neighbors' influence. Both of these models are related to the Delphi method ([4], [22]) developed at the RAND Corporation for consensus finding and problem solving in groups. Latané used neural networks to in [17], [18], and [19] to model social impact, and Merrill presented similar models in [24]. Falmagne and his colleagues in [6], [10], and [11] explored models for approval voting [2], where voters' opinions varied according to a stochastic stream of tokens or simple messages.

The idea of confidence levels was inspired by the work of Hoffman [15] and Roberts [27] who considered the use of confidence levels in deterministic models of opinion formulation on graphs. Confidence in models of opinion formulation is analogous to disease resistance in epidemiological models since both dynamically affect how quickly an individual can have his or her opinion changed, or be infected. Insights into the effect of confidence levels may lead to a greater understanding of how resistance affects epidemiological models.

In section 2.2 we provide a brief discussion of the original voter model to establish terminology and some of the tools that we will be using. In section 2.3 we present our version of the voter model with confidence levels and determine the limiting probability of entering the uniform 1 opinion, given the graph and the initial opinions. In section 2.4 we present some sample waiting functions and calculate the limiting probability for these functions on a sample graph. Section 2.5 mentions open questions and potential alternative ways to bring confidence levels into the voter model.

2.2 The Voter Model

The voter model is well studied, though often many authors focus on infinite graphs. We present here the voter model on arbitrary finite graphs, using our own terminology and notation. Let G be a finite, connected, undirected graph on n vertices labeled v_1

through v_n. We define a continuous-time Markov process $\{Z_t\}_{t\geq 0}$ on the finite state space $\{0,1\}^{V(G)}$ as follows. At every time $t \geq 0$, each vertex v_i has an *opinion* $Z_t(v_i)$ in $\{0,1\}$. Each vertex v_i asynchronously updates its opinion at times $\{T_\ell(v_i)\}_{\ell=1}^{\infty}$, where the sequence $\{T_\ell(v_i)\}$ is a Poisson process with rate 1; that is, $T_{\ell+1}(v_i) - T_\ell(v_i)$ is independently exponentially distributed with mean 1 for each ℓ. The $\{T_\ell(v_i)\}$ are also independent for each v_i. We set $T_0(v_i) = 0$ for convenience.

Vertex v_i updates its opinion at the times $\{T_\ell(v_i)\}_{\ell=1}^{\infty}$ according to the following rule: for $T_\ell(v_i) < t \leq T_{\ell+1}(v_i)$,

$$Z_t(v_i) = \begin{cases} 0, & \text{with probability } \#\{x \in N_G(v_i) : Z_{T_\ell(v_i)}(x) = 0\}/\#N_G(v_i), \\ 1, & \text{with probability } \#\{x \in N_G(v_i) : Z_{T_\ell(v_i)}(x) = 1\}/\#N_G(v_i). \end{cases}$$

Note that $Z_t(v_i)$ is updated only at the times $\{T_\ell(v_i)\}$[1], and is constant on each interval $(T_\ell(v_i), T_{\ell+1}(v_i)]$. Also observe that the update mechanism is probabilistically the same as v_i adopting the opinion of a neighbor $N_\ell(v_i)$ chosen uniformly at random from $N_G(v_i)$: for $T_\ell(v_i) < t \leq T_{\ell+1}(v_i)$,

$$Z_t(v_i) = Z_{T_\ell(v_i)}(N_\ell(v_i)), \text{ where } N_\ell(v_i) \text{ is chosen uniformly from } N_G(v_i).$$

We will use this view of the update mechanism for analyzing the voter model.

Because G is finite and connected, the only absorbing states for the process $\{Z_t\}_{t\geq 0}$ are the states of uniform opinion where either $Z_t(v_i) = 0$ for all i, or $Z_t(v_i) = 1$ for all i. Since the process will eventually enter one of these absorbing states with probability 1, we wish to know the probability of entering the uniform 1 opinion given the initial configuration Z_0 of opinions. All vertices have the same opinion in the absorbing states, and so this probability is the same as the limiting probability of v_i's opinion being 1, for any vertex v_i. Thus,

$$\Pr\{Z_t \text{ enters the uniform 1 opinion state} \mid Z_0\} = \lim_{t\to\infty} \Pr\{Z_t(v_i) = 1 \mid Z_0\}.$$

This probability can be found by considering another continuous-time Markov process $\{X_s\}$ that is dual to $\{Z_t\}$. The process $\{X_s\}_{s\geq 0}$ is a system of random walks, with one

[1]Strictly speaking, this isn't true since v_i's opinion does not change until immediately *after* $T_\ell(v_i)$. However, we will stick to the terminology that "v_i updates at $T_\ell(v_i)$."

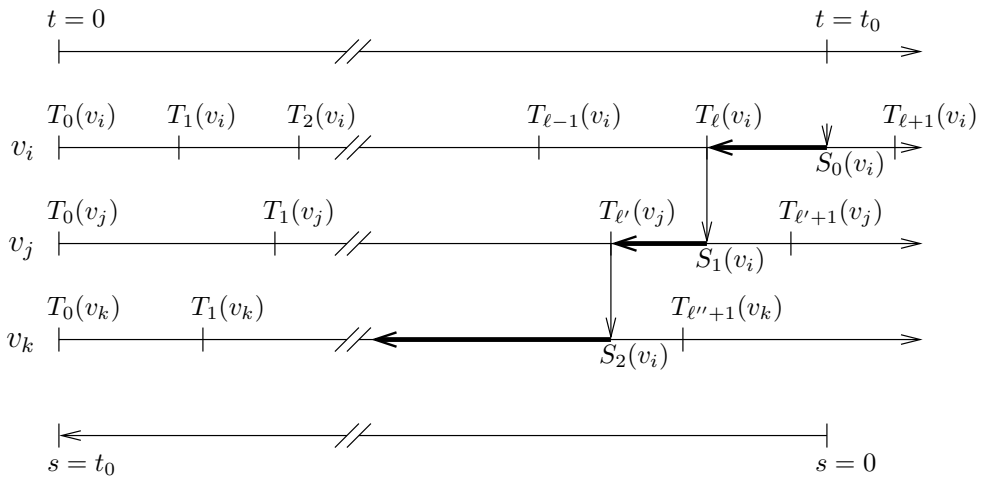

Figure 2.1: $\{X_s\}$ progresses backwards in time relative to $\{Z_t\}$.

random walk $X_s(v_i)$ starting at each vertex v_i. Formally, the process $\{X_s\}$ is defined on the finite state space $V(G)^{V(G)}$, where at every $s \geq 0$, each vertex has as its state $X_s(v_i)$ a vertex in $V(G)$. The random walk $X_s(v_i)$ updates at the discrete times $\{S_m(v_i)\}_{m=0}^{\infty}$. We think of the dual process as progressing *backwards* in time relative to $\{Z_t\}$, as shown in Figure 2.1. Fix a time $t_0 > 0$. The time $t = t_0$ for $\{Z_t\}$ corresponds to the time $s = 0$ for $\{X_s\}$. For each vertex v_i, $X_s(v_i)$ traces where v_i's opinion at time t_0 came from. We start with $X_0(v_i) = v_i$ and $S_0(v_i) = 0$. At some time $T_\ell(v_i)$, where $T_\ell(v_i) < t_0 \leq T_{\ell+1}(v_i)$, v_i adopted the opinion of a neighbor $v_j := N_\ell(v_i)$. We set $S_1(v_i) := t_0 - T_\ell(v_i)$ and update $X_{S_1(v_i)}(v_i)$ to v_j. For those s such that $0 \leq s < S_1(v_i)$, $X_s(v_i)$ is constant. At some time $T_{\ell'}(v_j)$, where $T_{\ell'}(v_j) < T_\ell(v_i) \leq T_{\ell'+1}(v_j)$, v_j adopted the opinion of a neighbor $v_k := N_{\ell'}(v_j)$. We set $S_2(v_i) := t_0 - T_{\ell'}(v_j)$ and update $X_{S_2(v_i)}(v_i) = v_k$. For those s such that $S_1(v_i) \leq s < S_2(v_i)$, $X_s(v_i)$ is constant. We continue this process, following v_i's opinion backwards in time, for $s \leq t_0$. Since v_i's opinion at time t_0 is the opinion of $X_s(v_i)$ at time $t_0 - s$, we have that

$$Z_{t_0}(v_i) = Z_{t_0-s}(X_s(v_i)).$$

Since these random variables are equal, we also have

$$\Pr\{Z_{t_0}(v_i) = 1 \mid Z_0\} = \Pr\{Z_{t_0-s}(X_s(v_i)) = 1 \mid Z_0\}.$$

By setting $s = t_0$ and taking the limit as $t_0 \to \infty$, we conclude that

$$\lim_{t_0 \to \infty} \Pr\{Z_{t_0}(v_i) = 1 \mid Z_0\} = \lim_{t_0 \to \infty} \Pr\{Z_0(X_{t_0}(v_i)) = 1 \mid Z_0\}. \tag{2.1}$$

The left hand limit in equation (2.1) is the quantity we want, and thus we have reduced the problem to calculating the limiting distribution of $X_s(v_i)$. The process $X_s(v_i)$ is simply a random walk where $X_s(v_i)$ moves to a neighbor chosen uniformly at random at the discrete times $\{S_m(v_i)\}_{m=0}^{\infty}$. How is $\{S_m(v_i)\}$ distributed? Again, fix $t_0 > 0$, and let m be such that $S_{m+1}(v_i) < t_0$. Set $v_j := X_{S_m(v_i)}(v_i)$, and let ℓ be such that $T_\ell(v_j) < t_0 - S_m(v_i) \leq T_{\ell+1}(v_j)$. Note that $S_{m+1}(v_i) = t_0 - T_\ell(v_j)$. The length of the interval $T_{\ell+1}(v_j) - T_\ell(v_j)$ is exponentially distributed with mean 1, and indicates no information about the direction of time. Thus, the random variable $T_{\ell+1}(v_j) - T_\ell(v_j)$ conditioned on $T_\ell(v_j)$ and the random variable $T_{\ell+1}(v_j) - T_\ell(v_j)$ conditioned on $T_{\ell+1}(v_j)$ are both exponentially distributed with mean 1. Since exponential distributions are memoryless, the distribution of when all the random walks $X_s(v_i)$ move from a vertex v_j is independent of when the random walks arrived at v_j. Hence, $S_{m+1}(v_i) - S_m(v_i)$ is exponentially distributed with mean 1. Since the $\{T_{\ell+1}(v_j) - T_\ell(v_j)\}$ are independent for each v_j and ℓ, the $\{S_{m+1}(v_i) - S_m(v_i)\}$ are also independent for each v_i and m. Therefore, $\{X_s(v_i)\}_{s \geq 0}$ is a standard continuous-time random walk on G; that is, $\{X_s(v_i)\}_{s \geq 0}$ is a random walk that moves to a neighbor chosen uniformly at random according to a Poisson process with rate 1. From the theory of random walks (see, for example, [23]),

$$\lim_{s \to \infty} \Pr\{Z_0(X_s(v_i)) = 1\} = \frac{\displaystyle\sum_{x \in V(G): Z_0(x) = 1} \deg_G(x)}{\displaystyle\sum_{x \in V(G)} \deg_G(x)}. \tag{2.2}$$

Note again that the graph structure does not enter into equation (2.2) and the degrees of the vertices do. The theory of random walks actually gives us more precise information, in that we can calculate $\lim_{s \to \infty} \Pr\{X_s(v_i) = x\}$. Summing these values over all x such that $Z_0(x) = 1$ gives us equation (2.2) above.

The random processes $\{X_s(v_i)\}$ for different vertices v_i are coupled together in the following way. Suppose that $X_s(v_i) = X_s(v_j) = v_k$ for $T_\ell(v_k) < t_0 - s \leq T_{\ell+1}(v_k)$. Then $X_{t_0 - T_\ell(v_k)}(v_i) = X_{t_0 - T_\ell(v_k)}(v_j) = N_\ell(v_k)$. In words, if $X_s(v_i)$ and $X_s(v_j)$ are both at v_k,

then $X_s(v_i)$ and $X_s(v_j)$ move *together* to the vertex $N_\ell(v_k)$ that is chosen uniformly from v_k's neighborhood. From that point on, $X_s(v_i)$ and $X_s(v_j)$ remain coupled, in the sense that $X_s(v_i) = X_s(v_j)$ for $s \geq t_0 - T_\ell(v_j)$. The random walks $\{X_s\}$ are called *coalescing random walks*, and two random walks $X_s(v_i)$ and $X_s(v_j)$ are said to *coalesce* when the two random walks hit and move together thereafter. With probability 1, all of the random walks coalesce (see [20]); that is, there is some time S such that $X_s(v_i) = X_s(v_j)$ for $s > S$ and for all v_i and v_j. This provides another proof of the fact that the only absorbing states for the process $\{Z_t\}_{t\geq 0}$ are the states of uniform opinion.

2.3 The Voter Model with Confidence Levels

We now extend the voter model to include the *confidence* a voter has in its opinion. Specifically, we include the idea that a voter is less likely to consider the opinions of neighbors the higher its confidence level is. We present one specific set of assumptions for this extension, and in section 2.5 we point out other possibilities for the assumptions for future consideration.

To motivate this extension, we consider a group of voters in a two-party political system. All of the relationships between voters are known and are represented as edges in the graph G. We assume that the two parties have opposing views on some issue, and that the members of each party initially assume their party's position by default before discussion begins. As the discussion progresses, a voter v_i is equally swayed by all of its neighbors in G, whether they are in its party or not. However, members of the same party are trusted associates in the sense that if trusted associate v_j convinces v_i of its opinion, then v_j also convinces v_i of v_j's confidence level in that opinion. In fact, v_i's new confidence level is even one greater than v_j's confidence level $C_t(v_j)$. Not only does v_i accept v_j's confidence in the opinion, but v_j's own credibility is added to v_i's confidence level. When v_i adopts the opinion of a non-trusted associate, then its confidence level simply resets to a level of 0. The effect of a higher confidence level is that it takes longer for any of v_i's neighbors to convince v_i of a new opinion.

We now formally specify the voter model with confidence levels. As above, we have a

finite connected undirected graph G and a continuous-time stochastic process $\{(Z_t, C_t)\}_{t\geq 0}$ defined on the state space $(\{0,1\} \times \mathbb{N})^{V(G)}$. At every time $t \geq 0$, each vertex v_i has an *opinion* $Z_t(v_i)$ in $\{0,1\}$ and a *confidence level* $C_t(v_i)$ in $\mathbb{N}$. We use the confidence level to count, in some sense, the number of times that v_i's current opinion has been affirmed by other voters. A vertex v_i asynchronously updates its opinion at times $\{T_\ell(v_i)\}_{\ell=1}^{\infty}$ by adopting the opinion of a neighbor $v_j := N_\ell(v_i)$ chosen uniformly at random; that is, for $T_\ell(v_i) < t \leq T_{\ell+1}(v_i)$,

$$Z_t(v_i) = Z_{T_\ell(v_i)}(v_j), \text{ where } v_j := N_\ell(v_i) \text{ is chosen uniformly from } N_G(v_i).$$

Vertex v_i's confidence level is updated at the same times $\{T_\ell(v_i)\}$. However, these times are not distributed as before, so we will specify their distribution below. To specify how the confidence level is updated, we define v_j to be a *trusted associate* of v_i if and only if v_j's initial opinion $Z_0(v_j)$ agrees with v_i's initial opinion $Z_0(v_i)$. The *trusted component* $K(v_i)$ of v_i is the component containing v_i of the subgraph of G induced by vertices whose initial opinions agree with v_i's. Note that while opinions and confidence levels change as the process progresses, the trusted relationship does not. For $T_\ell(v_i) < t \leq T_{\ell+1}(v_i)$, $C_t(v_i)$ is updated as follows:

$$C_t(v_i) = \begin{cases} C_{T_\ell(v_i)}(v_j) + 1 & \text{if } v_j = N_\ell(v_i) \text{ is a trusted associate of } v_i, \\ 0 & \text{if not.} \end{cases} \tag{2.3}$$

Notice that v_i increments v_j's confidence level if $v_j = N_\ell(v_i)$ is a trusted associate, and not v_i's own confidence level.

Rather than specifying the initial confidence level $C_0(v_i)$ as a single value, we instead define $C_0(v_i)$ to be a random variable with distribution

$$\Pr\{C_0(v_i) = \ell\} = \Pr\{X_s(v_i) \text{ leaves } K(v_i) \text{ in exactly } \ell \text{ steps}\}.$$

We will explain this assumption during our analysis of the voter model, particularly what the process $X_s(v_i)$ is. However, the reason for this choice is that $C_0(v_i)$ is a "typical" value representative of the values that occur as the process $\{(Z_t, C_t)\}$ is progressing. This assumption simplifies the analysis since the initial condition for $C_t(v_i)$ looks

the same as when the process is running. When computing the limiting probability $\lim_{t\to\infty} \Pr\{Z_0(X_t(v_i)) = 1 \mid Z_0\}$, we condition on Z_0 (and intrinsically on G), but *not* on C_0.

The length of the time interval $(T_\ell(v_i), T_{\ell+1}(v_i)]$ is dependent purely on the confidence level $C_t(v_i)$ that v_i has during that time interval. In most cases, we want a higher confidence level to mean a longer time before opinions are updated. Let the *waiting function* $f : \mathbb{N} \to \mathbb{R}^+$ be a positive function whose differences $f(c) - f(c-1)$ are increasing for positive integers c. If $c = C_{T_\ell(v_i)}(v_i)$, then $T_{\ell+1}(v_i) - T_\ell(v_i)$ is exponentially distributed with mean $f(c) - f(c-1)$ if c is positive, and exponentially distributed with mean $f(0)$ if $c = 0$. We set $T_0(v_i) = 0$ for convenience. This choice of how confidence levels affects update times is again chosen to simplify the analysis of the voter model. However, it is flexible enough to consider several different waiting functions f, and we consider several examples in section 2.4.

To analyze $\{(Z_t, C_t)\}$, we form the dual process $\{(X_s, D_s)\}_{s\geq 0}$, where, as described above, the dual is progressing *backwards* in time relative to $\{(Z_t, C_t)\}$. Fix a time $t_0 > 0$. Again, $\{X_s(v_i)\}_{s\geq 0}^{s\leq t_0}$ traces where v_i's opinion at time t_0 came from, and so we have

$$Z_{t_0}(v_i) = Z_{t_0-s}(X_s(v_i)), \text{ and}$$
$$Z_{t_0}(v_i) = Z_0(X_{t_0}(v_i)), \text{ by setting } s = t_0.$$

The process $\{X_s(v_i)\}_{s\geq 0}$ is a random walk on G that moves to a neighbor $N_m(v_i)$ chosen uniformly at random at the discrete times $\{S_m(v_i)\}_{m=0}^{\infty}$, where $S_0(v_i)$ is defined to be 0. In keeping with our view of the dual progressing backwards in time, the variable $D_s(v_i)$ is defined to be the confidence level $C_{t_0-s}(X_s(v_i))$ of the vertex $X_s(v_i)$. The distribution of $S_{m+1}(v_i) - S_m(v_i)$ is dependent on $c := D_s(v_i)$, for $S_m(v_i) \leq s < S_{m+1}(v_i)$. Note that c is constant for s in this interval. As before, since the exponential distribution is independent of the direction of time and is memoryless, $S_{m+1}(v_i) - S_m(v_i)$ is exponentially distributed with mean $f(c) - f(c-1)$ if c is positive, and exponentially distributed with mean $f(0)$ if $c = 0$.

Note that $D_s(v_i)$ updates only when $X_s(v_i)$ updates, and that when $X_s(v_i)$ does not leave the current trusted component $K(X_s(v_i))$, then $D_s(v_i)$ simply decrements by 1. In

fact, $X_s(v_i)$ does not leave the trusted component until $D_s(v_i) = 0$. The critical observation is that $D_s(v_i)$ thus counts the number of steps that $X_s(v_i)$ *will take* in the current trusted component before leaving. It is slightly disconcerting that $D_s(v_i)$ is dependent on the future (in the sense of the s time variable). However, since the random variable $N_m(v_i)$ of a randomly chosen neighbor at time $S_m(v_i)$ is independent of $N_{m'}(v_j)$ for any m' and $v_j \neq v_i$, of $N_{m'}(v_i)$ for any $m' \neq m$, and of $S_{m'}(v_j)$ for any m' and v_j, and since $D_s(v_i)$ is dependent only on the choices $\{N_m(v_i)\}$, we will be able to analyze the situation with only a little extra difficulty. This explains our choice of the distribution of $C_0(v_i)$:

$$\Pr\{C_0(v_i) = \ell\} = \Pr\{X_s(v_i) \text{ leaves } K(v_i) \text{ in exactly } \ell \text{ steps}\}.$$

The random walks $\{X_s\}$ are again coupled in that if $X_{s'}(v_i) = X_{s'}(v_j)$ for some s', then $X_s(v_i) = X_s(v_j)$ for all $s \geq s'$. The random variables D_s also are coupled since if $X_s(v_i) = X_s(v_j)$ for all $s \geq s'$, then so $D_s(v_i) = D_s(v_j)$ for all $s \geq s'$. The system of random walks $\{X_s\}$ do coalesce into a single random walk. However, this is not immediate, and will be proved in Theorem 2.3.

From the construction of the dual, we immediately have

Proposition 2.1 (Duality). *The process $\{(X_s, D_s)\}_{s\geq 0}$ is dual to $\{(Z_t, C_t)\}_{t\geq 0}$ in the sense that*

$$\Pr\{Z_{t_0}(v_i) = 1 \mid Z_0\} = \Pr\{Z_0(X_{t_0}(v_i)) = 1 \mid Z_0\}, \text{ and so}$$

$$\lim_{t_0\to\infty} \Pr\{Z_{t_0}(v_i) = 1 \mid Z_0\} = \lim_{s\to\infty} \Pr\{Z_0(X_s(v_i)) = 1 \mid Z_0\},$$

if at least one of the limits exists.

Definition 2.2. Let $\{\widehat{X}_m(v_i)\}_{m=0}^{\infty}$ be the discrete-time Markov chain given by $\widehat{X}_m(v_i) = X_{S_m(v_i)}(v_i)$. Known as the *embedded chain* or *jump chain* for the process $\{(X_s(v_i), D_s(v_i)\}_{s\geq 0}$, $\{\widehat{X}_m(v_i)\}$ is defined by the state transition probabilities without respect to the holding times at each state. In our case, $\{\widehat{X}_m(v_i)\}$ is a standard discrete-time random walk on G; that is, $\{\widehat{X}_m(v_i)\}_{m=0}^{\infty}$ is a random walk that moves to a neighbor chosen uniformly at random at each discrete time unit.

Theorem 2.3 (Coalescence). *If Z_0 is not a uniform opinion state, then the system $\{(X_s, D_s)\}_{s\geq 0}$ of random walks almost surely coalesces to a single random walk; that is, with probability 1 there exists a time S_c such that for all $s \geq S_c$ and any v_i and v_j, $X_s(v_i) = X_s(v_j)$.*

Proof. For the sake of clarity and simplicity of notation, we will refer to the random variables $X_s(v_i)$, $X_s(v_j)$, and $\widehat{X}_m(v_i)$ as X_s, X'_s, and $\widehat{X}_m$, respectively. For the rest of the proof, v_i and v_j will simply be arbitrary vertices, and not necessarily the starting vertices of the random walks.

If G is connected and non-bipartite, then the state space of $\{\widehat{X}_m\}$ contains cycles of both even and odd length. By Corollary 1 of section 5.6 of [26], $\{\widehat{X}_m\}$ is a regular Markov chain. Thus, there exists an integer M such that

$$\Pr\{\widehat{X}_m = v_k \mid \widehat{X}_0 = v_i\} > \varepsilon \tag{2.4}$$

for any $m \geq M$, any v_i and v_k, and for some positive constant ε. Let $\mathcal{E}_{i,k,s'}$ be the event that the number of updates of X_s (or "steps" of $\widehat{X}_m$) that occur between $s = 0$ and $s = s'$ is at least M, given that $X_0 = v_i$ and that $X_{s'} = v_k$. Because Z_0 is not a uniform opinion state, $K(v_i)$ is not all of G. Thus, there exists a sequence of at most $2M$ steps of $\widehat{X}_m$ from v_i to v_k to some vertex $v_j \notin K(v_k)$ such that the sequence leaves $K(v_i)$. If v_i and v_k are in the same trusted component, then one possibility is that the sequence starts at v_i, visits v_k, and then exits $K(v_i)$ to v_j. If v_i and v_k are in different trusted components, then the sequence could start at v_i, leave $K(v_i)$ to visit v_k, and then exit $K(v_i)$ to v_j. Note that v_j is the first vertex visited when $\widehat{X}_m$ exits $K(v_k)$. Since the sequence leaves both $K(v_i)$ and $K(v_k)$, the confidence level at each step is bounded by $2M$, and so the time to take each of these steps has finite expectation bounded by $f(2M) - f(2M-1)$. Thus, there exists a time $S_{i,k}$ and a positive constant $\delta_{i,k}$, both dependent on v_i and v_k, such that the probability of $\mathcal{E}_{i,k,S_{i,k}}$ is greater than $\delta_{i,k}$. Let S be the maximum of $S_{i,k}$ over all i and k, and similarly let δ be the minimum of $\delta_{i,k}$ over all i and k. Then $\mathcal{E}_{i,k,S}$

has probability greater than δ, for any choice of v_i and v_k. Thus,

$$\begin{aligned}
&\Pr\{X_S = v_k \mid X_0 = v_i\} \\
&= \sum_{m=0}^{\infty} \Pr\{X_s \text{ takes } m \text{ steps during } s \in (0, S]\} \Pr\{\widehat{X}_m = v_k \mid \widehat{X}_0 = v_i\} \\
&\geq \sum_{m=M}^{\infty} \Pr\{X_s \text{ takes } m \text{ steps during } s \in (0, S]\} \Pr\{\widehat{X}_m = v_k \mid \widehat{X}_0 = v_i\} \\
&> \sum_{m=M}^{\infty} \Pr\{X_s \text{ takes } m \text{ steps during } s \in (0, S]\}\varepsilon \qquad \text{by (2.4)}, \\
&> \delta\varepsilon,
\end{aligned}$$

and so

$$\Pr\{X_S = v_k \mid X_0 = v_i\} > \delta\varepsilon. \tag{2.5}$$

If G is bipartite, we need to consider the parity of the number of steps taken. By Theorem 5.13 of [26], $\{\widehat{X}_m\}$ is an ergodic Markov chain of period two. Thus, there exists an integer M such that

$$v_i, v_k \text{ in the same partite set} \Rightarrow \Pr\{\widehat{X}_m = v_k \mid \widehat{X}_0 = v_i\} > \varepsilon \quad \text{if } m \text{ is even}, \tag{2.6}$$

and

$$v_i, v_k \text{ in different partite sets} \Rightarrow \Pr\{\widehat{X}_m = v_k \mid \widehat{X}_0 = v_i\} > \varepsilon \quad \text{if } m \text{ is odd}, \tag{2.7}$$

where $m \geq M$ and ε is some positive constant. Let $\mathcal{E}^{\mathrm{e}}_{i,k,s'}$ be the event that the number of updates of X_s (or "steps" of $\widehat{X}_m$) that occur between $s = 0$ and $s = s'$ is at least M and of the appropriate parity, given that $X_0 = v_i$ and that $X_{s'} = v_k$. Because Z_0 is not a uniform opinion state, $K(v_i)$ is not all of G. Thus, there exists a sequence of at most $2M+2$ steps of $\widehat{X}_m$ from v_i to v_k to some vertex $v_j \notin K(v_k)$ such that the sequence leaves $K(v_i)$. As before, if v_i and v_k are in the same trusted component, then one possibility is that the sequence starts at v_i, visits v_k, and then exits $K(v_i)$ to v_j. If v_i and v_k are in different trusted components, then the sequence could start at v_i, leave $K(v_i)$ to visit v_k, and then exit $K(v_i)$ to v_j. Note that v_j is the first vertex visited when $\widehat{X}_m$ exits $K(v_k)$. Since the sequence leaves both $K(v_i)$ and $K(v_k)$, the confidence level at each step is bounded by $2M + 2$, and so the time to take each of these steps has finite expectation

bounded by $f(2M+2) - f(2M+1)$. Hence, there exists an $S_{i,k}$ and a positive constant $\delta_{i,k}$, both dependent on v_i and v_k, such that the probability of $\mathcal{E}^{\mathrm{e}}_{i,k,S_{i,k}}$ is greater than $\delta_{i,k}$. Let S be the maximum of $S_{i,k}$ over all i and k, and similarly let δ be the minimum of $\delta_{i,k}$ over all i and k. Then $\mathcal{E}^{\mathrm{e}}_{i,k,S}$ has probability greater than δ, for any choice of v_i and v_k. Thus,

$$\begin{aligned}
&\Pr\{X_S = v_k \mid X_0 = v_i\} \\
&= \sum_{m=0,\ m \text{ even}}^{\infty} \Pr\{X_s \text{ takes } m \text{ steps during } s \in (0,S]\} \Pr\{\widehat{X}_m = v_k \mid \widehat{X}_0 = v_i\} \\
&\geq \sum_{m=M,\ m \text{ even}}^{\infty} \Pr\{X_s \text{ takes } m \text{ steps during } s \in (0,S]\} \Pr\{\widehat{X}_m = v_k \mid \widehat{X}_0 = v_i\} \\
&> \sum_{m=M,\ m \text{ even}}^{\infty} \Pr\{X_s \text{ takes } m \text{ steps during } s \in (0,S]\}\varepsilon \qquad \text{by (2.6)}, \\
&> \delta\varepsilon,
\end{aligned}$$

and so

$$\Pr\{X_S = v_k \mid X_0 = v_i\} > \delta\varepsilon. \tag{2.8}$$

Similarly, if v_i and v_k are in different partite sets, then the same proof works by considering only an odd number of steps.

Note that the bounds (2.5), (2.8), and the analogous bound when v_i and v_k are in different partite sets are time translation invariant, in the sense that for any $s > 0$,

$$\Pr\{X_{S+s} = v_k \mid X_s = v_i\} > \delta\varepsilon.$$

This is because X_s is independent of the past.

We now consider a second random walk $\{X'_s\}$. We wish to determine the probability of X_s and X'_s hitting; that is, of there existing a time S_c such that $X_{S_c} = X'_{S_c}$. Let S_h be the first time when $X_s = X'_s$ if X_s and X'_s hit, and let $S_h = \infty$ if X_s and X'_s never hit. Given that $X_0 = v_i$ and $X'_0 = v_j$, the probability that S_h is at most the parameter S from above is exactly equal to the probability that X_S and X'_S both equal a vertex v_k,

for some vertex v_k. Thus, we have

$$\begin{aligned}\Pr\{S_h \le S \mid X_0 = v_i, X'_0 = v_j\} &= \sum_{k=1}^{n} \Pr\{X_S = v_k \mid X_0 = v_i\} \Pr\{X'_S = v_k \mid X'_0 = v_j\} \\ &> \sum_{k=1}^{n} (\delta\varepsilon) \Pr\{X'_S = v_k \mid X'_0 = v_j\} \qquad \text{by (2.5) and (2.8)}, \\ &= \delta\varepsilon.\end{aligned}$$

This result is also time translation invariant, in that

$$\Pr\{S_h \le S + s \mid X_s = v_i, X'_s = v_j\} > \delta\epsilon$$

for any $s > 0$.

We now consider what happens in q consecutive time intervals each of length S. The probability that X_s and X'_s do not hit in any of the intervals is at most $(1 - \delta\varepsilon)^q$, and so

$$\Pr\{S_c \le Sq \mid X_0 = v_i, X'_0 = v_j\} \ge 1 - (1 - \delta\varepsilon)^q \to 1 \text{ as } q \to \infty.$$

Thus, with probability 1, the random walks X_s and X'_s hit and so coalesce. Applying this result to all n random walks in the system $\{(X_s, D_s)\}$, we have that with probability 1, the system $\{(X_s, D_s)\}$ coalesces into a single random walk. □

Note that Theorem 2.3 trivially implies that $\{D_s\}_{s\ge 0}$ also coalesces, *i.e.*, that $D_s(v_i) = D_s(v_j)$ for all $s \ge S_c$. This is true because $D_s(v_i) = D_s(v_j)$ whenever $X_s(v_i) = X_s(v_j)$.

Since the system coalesces, we will now drop the vertex name and only write X_s, D_s, and $\widehat{X}_m$. We wish to calculate $\lim_{s\to\infty} \Pr\{Z_0(X_s) = 1 \mid Z_0\}$; what we will actually calculate is the more detailed $\lim_{s\to\infty} \Pr\{X_s \in K(v_i) \mid Z_0\}$ for vertex v_i. Note that because of our setup it would be very difficult to calculate $\lim_{s\to\infty} \Pr\{X_s = v_i \mid Z_0\}$ as is done in the original voter model.

To analyze the limiting probability distribution of X_s we create a new semi-Markov process $\widetilde{X}_s$ that captures the movement of X_s between different trusted components. Since this only makes sense when there are different trusted components, we henceforth assume that the initial set of opinions Z_0 is not uniform.

Definition 2.4. The continuous-time process $\{\widetilde{X}_s\}_{s\ge 0}$ is defined on the state space of vertices $v_1, \ldots, v_n$ of G, where we assume that Z_0 is not uniform. Given that $X_s \in K(v_i)$,

let $\widetilde{X}_s$ be the vertex v_j in $K(v_i)$ that X_s first visited when entering $K(v_i)$. Formally, let s' be the least time such that $X_{s''} \in K(v_i)$ for all $s' \leq s'' \leq s$. Then let $\widetilde{X}_s = X_{s'}$.

Since X_s is dependent only on its immediate future that is spent in the current trusted component before leaving, $\widetilde{X}_s$ is a semi-Markov process. The transition probability p_{ik} from v_i to v_k is the probability that X_s first exits $K(v_i)$ to v_k given that $X_0 = v_i$. Note that $p_{ik} = 0$ if $v_k \in K(v_i)$ or if v_k is not a neighbor to any vertex in $K(v_i)$. The holding time H_i of $\widetilde{X}_s$ at v_i is the time X_s takes to leave $K(v_i)$, given that $X_0 = v_i$. [2] Note also that the definition of C_0 as a random variable representing a typical value of C_t allows us to form the semi-Markov process without needing to condition on the initial condition C_0.

Lemma 2.5. *The process $\{\widetilde{X}_s\}_{s\geq 0}$ is dual to $\{X_s\}_{s\geq 0}$ in the sense that*

$$\Pr\{X_s \in K(v_i) \mid Z_0\} = \Pr\{\widetilde{X}_s \in K(v_i) \mid Z_0\}, \text{ and so}$$
$$\lim_{s\to\infty} \Pr\{X_s \in K(v_i) \mid Z_0\} = \lim_{s\to\infty} \Pr\{\widetilde{X}_s \in K(v_i) \mid Z_0\},$$

if at least one of the limits exists.

Proof. Note that if $X_s \in K(v_i)$, then $\widetilde{X}_s = v_j$ for some $v_j \in K(v_i)$. Then

$$\begin{aligned}
\Pr\{X_s \in K(v_i) \mid Z_0\} &= \Pr \bigcup_{v_j \in K(v_i)} \{X_s \in K(v_i) \text{ and } \widetilde{X}_s = v_j \mid Z_0\} \\
&= \Pr \bigcup_{v_j \in K(v_i)} \{\widetilde{X}_s = v_j \mid Z_0\} \qquad \text{since } \widetilde{X}_s = v_j \text{ implies } X_s \in K(v_i), \\
&= \Pr\{\widetilde{X}_s \in K(v_i) \mid Z_0\}.
\end{aligned}$$

The last statement of the proposition follows by taking the limit as $s \to \infty$ of both sides of the above equation. □

If we know the limiting probability of $\widetilde{X}_s$, then summing these probabilities over all

[2] Since X_s is independent of its past, the transition probability p_{ik} and the holding time H_i are the same if the definitions condition on $X_{s'} = v_i$ for some specific time s'. However, for convenience, we simply shift time so that $s' = 0$.

vertices with initial opinion 1, we have

$$\sum_{v_i:Z_0(v_i)=1} \lim_{s\to\infty} \Pr\{\widetilde{X}_s = v_i \mid Z_0\} = \sum_{v_i:Z_0(v_i)=1} \lim_{s\to\infty} \Pr\{X_s \in K(v_i) \mid Z_0\}$$
$$= \lim_{s\to\infty} \Pr\{Z_0(X_s) = 1 \mid Z_0\}$$

where the last expression is the quantity we are interested in. The theory of semi-Markov processes enables us to calculate the limiting probability distribution of $\widetilde{X}_s$, if the expected holding time $\mathrm{E}[H_i]$ is finite for each i (see [13]). Let $\{\widehat{\widetilde{X}}_r\}_{r=0}^{\infty}$ be the embedded Markov chain for $\widetilde{X}_s$. Then

$$\lim_{s\to\infty} \Pr\{\widetilde{X}_s \in K(v_i) \mid Z_0\} = \frac{\displaystyle\sum_{v_j\in K(v_i)} \left(\mathrm{E}[H_j] \lim_{r\to\infty} \Pr\{\widehat{\widetilde{X}}_r = v_j\}\right)}{\displaystyle\sum_{k=1}^{n} \mathrm{E}[H_k]}. \tag{2.9}$$

Note that $\{\widehat{\widetilde{X}}_r\}$ is defined by the transition probabilities p_{ik}, which in turn can be determined from the Markov chain $\{\widehat{X}_m\}$. Since $\{\widehat{X}_m\}$ ignores waiting times, $\{\widehat{X}_m\}$ and hence $\{\widehat{\widetilde{X}}_r\}$ is independent of the choice of the waiting function f. To explicitly calculate p_{ik}, let A denote the transition matrix of $\widehat{X}_m$ on G, and let A_H denote the restriction of A to a subgraph H of G. We denote the complement in G of a subgraph H by $\overline{H}$. Let $A_{H\times\overline{H}}$ denote the restriction of A to transitions from H to $\overline{H}$. Given that $X_0 = v_i$, we wish to compute the probability that X_s exits $K(v_i)$ to vertex $v_k \in \overline{K(v_i)}$. This probability is exactly the probability that, given that $\widehat{X}_0 = v_i$, $\widehat{X}_m$ leaves $K(v_i)$ by going to vertex v_k. Using Markov chain theory (see, for example, chapter 5 of [26]), the matrix

$$B = (I - A_{K(v_i)})^{-1} A_{K(v_i)\times\overline{K(v_i)}}$$

gives the exit probabilities of $\widehat{X}_m$ from $K(v_i)$. That is, if $B = [b_{jk}]$, then

$$b_{jk} = \Pr\{\widehat{X}_m \text{ exits } K(v_i) \text{ to } v_k \in \overline{K(v_i)} \mid \widehat{X}_0 = v_j \in K(v_i)\}.$$

We also need to calculate the expected holding time $\mathrm{E}[H_i]$, which is just the expected time s when X_s exits $K(v_i)$, given that $X_0 = v_i$. Recall that the ik^{th} entry of $A^d_{K(v_i)} A_{K(v_i)\times\overline{K(v_i)}}$ gives the probability that $\widehat{X}_m$ exits $K(v_i)$ to $v_k \in \overline{K(v_i)}$ in exactly

$d+1$ steps, given that $\widehat{X}_0 = v_i$. Thus,

$$\begin{aligned}
&\text{E[time for } X_s \text{ to exit } K(v_i) \text{ to } v_k \mid X_0 = v_i \text{ and } X_s \text{ exits to } v_k] \\
&= \sum_{d=0}^{\infty} \Big(\text{E[time for } X_s \text{ to exit } K(v_i) \mid \widehat{X}_m \text{ exits to } v_k \text{ in } d+1 \text{ steps and } \widehat{X}_0 = v_i] \times \\
&\qquad \Pr\{\widehat{X}_m \text{ exits to } v_k \text{ in } d+1 \text{ steps} \mid \widehat{X}_0 = v_i\} \Big)
\end{aligned}$$

The value of D_s determines the distribution of the holding time of X_s at v_i: D_s is the number of steps that will be taken in $K(v_i)$ before exiting $K(v_i)$. Let $v_i = u_0, u_1, \ldots, u_{d+1} = v_k$ be the path taken by $\widehat{X}_m$ in $K(v_i)$ before exiting $K(v_i)$ to v_k. The holding time of X_s at v_i is thus independent of v_i, v_k, and the sequence $u_1, \ldots, u_d$ of steps taken within $K(v_i)$ from v_i to v_k. Thus,

$$\begin{aligned}
&\text{E[time for } X_s \text{ to exit } K(v_i) \mid \widehat{X}_m \text{ exits in } d+1 \text{ steps to } v_k \text{ and } \widehat{X}_0 = v_i] \\
&= \text{E[time for } X_s \text{ to exit } K(v_i) \mid \widehat{X}_m \text{ exits in } d+1 \text{ steps}] \\
&= \sum_{q=0}^{d} \text{E[holding time for } X_s \text{ at } u_q \mid \widehat{X}_m \text{ exits } K(v_i) \text{ in } d+1-q \text{ steps } (\textit{i.e.}, D_s = d+1-q)] \\
&= f(0) + [f(1) - f(0)] + [f(2) - f(1)] + \ldots + [f(d) - f(d-1)] \\
&= f(d),
\end{aligned}$$

where the second to last equality follows from how the confidence level and f affects the time between updates. Thus, the expected time until X_s exits $K(v_i)$, given that $X_0 = v_i$ and that $X_s = v_k$, is

$$\begin{aligned}
&\text{E[time for } X_s \text{ to exit } K(v_i) \mid \widehat{X}_m \text{ exits to } v_k \text{ and } \widehat{X}_0 = v_i] \\
&= \sum_{d=0}^{\infty} \text{E[time for } X_s \text{ to exit } K(v_i) \mid \widehat{X}_m \text{ exits in } d+1 \text{ steps to } v_k \text{ and } \widehat{X}_0 = v_i] \times \\
&\qquad \Pr[X_s \text{ exits } K(v_i) \text{ in } d+1 \text{ steps} \mid \widehat{X}_m \text{ exits to } v_k \text{ and } \widehat{X}_0 = v_i] \\
&= \sum_{d=0}^{\infty} f(d) \frac{\Pr[X_s \text{ exits } K(v_i) \text{ in } d+1 \text{ steps to } v_k \mid \widehat{X}_0 = v_i]}{\Pr[\widehat{X}_m \text{ exits to } v_k \mid \widehat{X}_0 = v_i]} \\
&= \frac{1}{b_{ik}} \left(ik^{\text{th}} \text{ entry of } \sum_{d=0}^{\infty} f(d) A^d_{K(v_i)} A_{K(v_i) \times \overline{K(v_i)}} \right).
\end{aligned}$$

If the series is convergent, then the expected time is finite, and formula (2.9) for the

limiting distribution of $\widetilde{X}_s$ holds. In the next section, we will consider specific choices for the waiting function f and examine how the process $\{(X_s, D_s)\}$ behaves with each choice.

2.4 Specific Choices for the Waiting Function f

We first need a standard result about power series of matrices.

Lemma 2.6. *Let $\sum_{d=0}^{\infty} f(d)x^d$ be a power series in the complex variable x with radius of convergence R. Then $\sum_{d=0}^{\infty} f(d)M^d$ is a convergent power series in the matrix M if all of the eigenvalues of M have modulus less than R. Furthermore, if $r(x)$ is a rational function in x such that $r(x) = \sum_{d=0}^{\infty} f(d)x^d$ for all $|x| < R$, then $r(M) = \sum_{d=0}^{\infty} f(d)M^d$ if all of the eigenvalues of M have modulus less than R.*

This result follows from the fact that a power series formula r clearly holds for diagonalizable matrices, and the diagonalizable matrices are dense in the space of all matrices.

Definition 2.7. Let G^1 denote the subgraph of G induced by vertices v_i whose initial opinions $Z_0(v_i)$ are 1, and similarly define G^0. Let A denote the transition matrix of $\widehat{X}_m$ on G, and let A_H denote the restriction of A to a subgraph H of G. For each connected component K of G^1 or G^0, let $e(K)$ denote the largest modulus of eigenvalues of A_K. Define $\mu(G, Z_0)$ to be the maximum $e(K)$ taken over all components K of G^1 and G^0. If every component that attains the maximum $\mu(G, Z_0)$ has the same initial opinion, then define $\mu'(G, Z_0)$ to be the maximum $e(K)$ taken over all components K that have the opposite opinion.

Lemma 2.8. *For any non-uniform set of initial opinions Z_0, $\mu(G, Z_0) < 1$.*

Proof. This statement follows immediately from the fact that $\widehat{X}_m$ will leave each trusted component with probability 1. □

We now consider several specific waiting functions, and show how the voter model with confidence levels behaves with these choices. To make the examples concrete, we calculate $\lim_{s\to\infty} \Pr\{Z_t(v_i) = 1 \mid Z_0\}$ for the graph G shown in Figure 2.2 and for various parameters of the waiting functions. In the figure, vertices with initial opinion $Z_0(v_i) = 1$

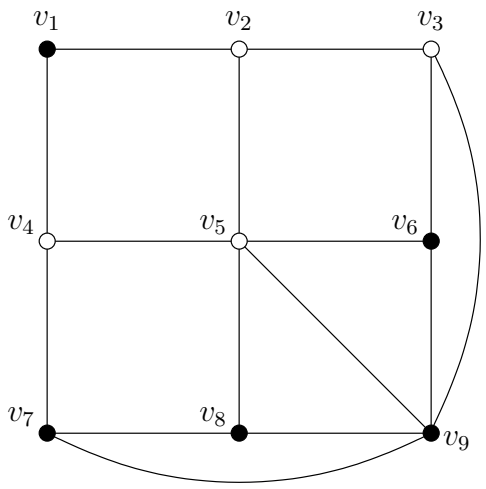

Figure 2.2: Initial opinions Z_0 on G. Vertices with initial opinion $Z_0(v_i) = 1$ are marked with black dots, while vertices with initial opinion 0 are marked with hollow dots. We refer to the three trusted components as $K(v_1)$, $K(v_2)$, and $K(v_6)$.

are marked with black dots, while vertices with initial opinion 0 are marked with hollow dots. We will refer to the three trusted components as $K(v_1)$, $K(v_2)$, and $K(v_6)$. The transition matrix A of $\widehat{X}_m$ on G is

$$A = \begin{bmatrix} 0 & 1/2 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 \\ 1/3 & 0 & 1/3 & 0 & 1/3 & 0 & 0 & 0 & 0 \\ 0 & 1/3 & 0 & 0 & 0 & 1/3 & 0 & 0 & 1/3 \\ 1/3 & 0 & 0 & 0 & 1/3 & 0 & 1/3 & 0 & 0 \\ 0 & 1/5 & 0 & 1/5 & 0 & 1/5 & 0 & 1/5 & 1/5 \\ 0 & 0 & 1/3 & 0 & 1/3 & 0 & 0 & 0 & 1/3 \\ 0 & 0 & 0 & 1/3 & 0 & 0 & 0 & 1/3 & 1/3 \\ 0 & 0 & 0 & 0 & 1/3 & 0 & 1/3 & 0 & 1/3 \\ 0 & 0 & 1/5 & 0 & 1/5 & 1/5 & 1/5 & 1/5 & 0 \end{bmatrix}.$$

It is interesting to note the behavior of the classic voter model, where $\lim_{t\to\infty} \Pr\{Z_t(v_i) = 1 \mid Z_0\} = 8/15 \approx .5333$ from equation (2.2).

Lemma 2.6 enables us to calculate the limiting probability for any polynomial waiting function. We present one specific class of polynomials here.

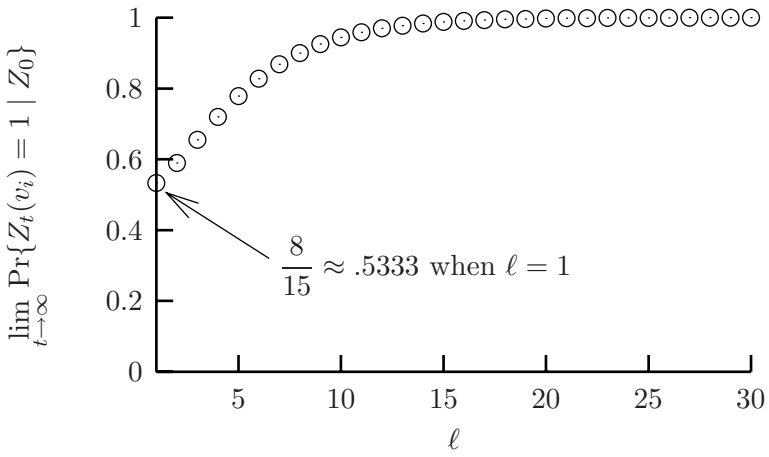

Figure 2.3: $\Pr\{Z_t$ enters the uniform 1 opinion state $\mid Z_0\}$ as a function of ℓ.

Example 2.9. Let f be the polynomial $f(d) = \binom{d+\ell}{\ell}$, where ℓ is a positive integer. Note that

$$\sum_{d=0}^{\infty} \binom{d+\ell}{\ell} x^d = \frac{1}{(1-x)^{\ell+1}}$$

by integrating the power series ℓ times. This series is convergent for $|x| < 1$, and so $\sum_{d=0}^{\infty} \binom{d+\ell}{\ell} A_{K(v_i)}^d = (I - A_{K(v_i)})^{-(\ell+1)}$ holds for any component $K(v_i)$. Figure 2.3 shows the limiting distribution for $\ell = 1, \ldots, 30$ for our example graph G. Note that when $\ell = 1$, $f(d) = d+1$, and the voter model with confidence levels reduces to the special case of the classic voter model. Thus, at $\ell = 1$, $\lim_{s\to\infty} \Pr\{Z_t(v_i) = 1 \mid Z_0\} = 8/15$.

Example 2.10. Let f be the exponential function $f(d) = \lambda^d$, where $\lambda > 1$. If $\lambda < 1/\mu(G, Z_0)$, then all of the eigenvalues of $\lambda A_{K(v_i)}$ are less than 1 in modulus for any component $K(v_i)$. Thus, $\sum_{d=0}^{\infty} \lambda^d A_{K(v_i)}^d$ is a convergent power series, and, using the formula for a convergent geometric series, is equal to $(I - \lambda A_{K(v_i)})^{-1} A_{K(v_i)\times\overline{K(v_i)}}$. For $\lambda \geq 1/\mu(G, Z_0)$, it is not clear that the limiting probability exists, since the theory of semi-Markov processes in general does not deal with infinite expected holding times. It is an open problem to determine the behavior beyond the threshold $1/\mu(G, Z_0)$.

When $\lambda = 1$, $f(d)$ is a constant function. Thus, the expected time for X_s to leave a trusted component is 1, regardless of the number of steps taken. The limiting probability

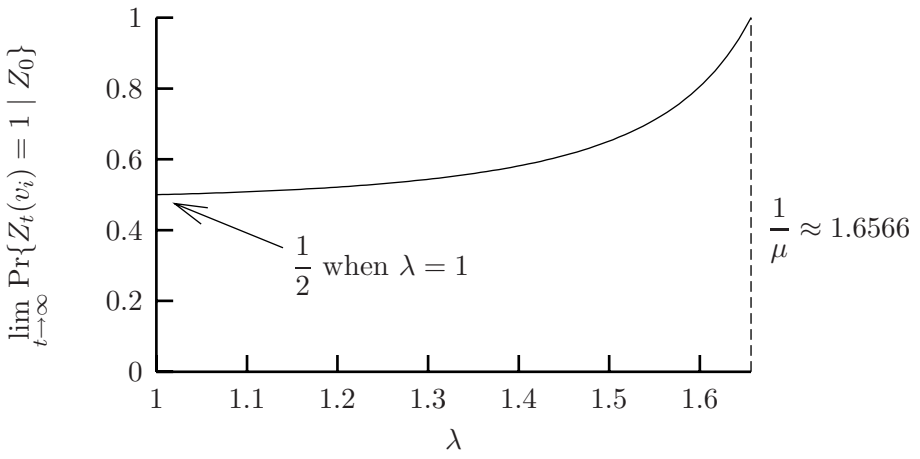

Figure 2.4: $\Pr\{Z_t$ enters the uniform 1 opinion state $\mid Z_0\}$ as a function of λ.

reduces to the stationary distribution of the embedded Markov chain $\widehat{\widetilde{X}}_r$ of the semi-Markov process $\widetilde{X}_s$. Since there are two opinions, the underlying graph of $\widehat{\widetilde{X}}_r$ is always a bipartite graph, and hence $\lim_{t\to\infty} \Pr\{Z_t(v_i) = 1 \mid Z_0\} = 1/2$.

For our example graph G, the largest eigenvalue in modulus of $A_{K(v_1)}$, $A_{K(v_2)}$, and $A_{K(v_6)}$ is $\mu(G, Z_0) \approx .6037$ from $A_{K(v_6)}$. We can thus calculate $\lim_{s\to\infty} \Pr\{Z_t(v_i) = 1 \mid Z_0\}$ for $1 < \lambda < 1/\mu(G, Z_0) \approx 1.6566$ using the geometric series formula.

We can also create a waiting function that is increasing with d yet bounded.

Example 2.11. Let f be the function $f(d) = 2-\theta^d$, for $0 < \theta < 1$. Here 2 is an arbitrary constant chosen for concreteness. Since

$$\sum_{d=0}^{\infty}(2 - \theta^d)x^d = \frac{2}{1-x} - \frac{1}{1-\theta x}$$

for $|x| < 1$, $\sum_{d=0}^{\infty}(2 - \theta^d)A^d_{K(v_i)}$ is convergent for any component $K(v_i)$. The limiting probability as a function of θ is shown in Figure 2.5. When θ is 0 or 1, $f(d)$ is constant, and, as seen above, the limiting probability is 1/2. Interestingly, the limiting probability is less than 1/2 for $0 < \theta < 1$, attaining a minimum at $\theta \approx .6430$. However, because the waiting function is bounded, high confidence levels do not significantly affect the limiting probability, and the limiting probability varies little for θ in the range $0 < \theta < 1$.

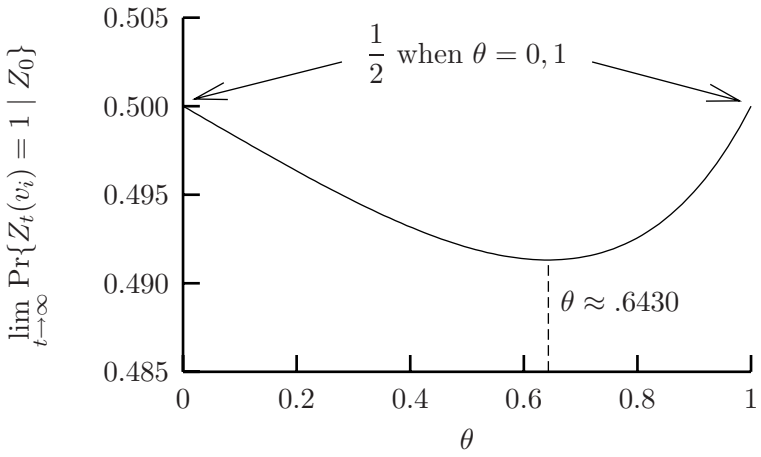

Figure 2.5: $\Pr\{Z_t$ enters the uniform 1 opinion state $\mid Z_0\}$ as a function of θ.

2.5 Conclusion

As with the classic voter model, the results we have presented on the voter model with confidence levels hold in slightly more general situations. More opinions than 0 and 1 are allowed, and the influence of a neighbor can be more general. If we allow G to be a digraph, have loops, and have weights on the edges, then we might modify the model so that when v_i updates, v_i randomly picks a neighbor v_j from $N_G^{\text{in}}(v_i)$ with probability proportional to the weight of the directed edge (v_j, v_i). In the digraph case, the digraph must be strongly connected for coalescence to occur, and the proof of Theorem 2.3 also needs to be strengthened in a straightforward way to handle larger periods in Markov chains. Also, trusted components can be any connected induced subgraphs as long as the initial opinions within the trusted components are uniform. Unfortunately, if certain features of the model are weakened, then the type of arguments we have made no longer work. The independence of the confidence level and the opinion held is absolutely crucial for the dual to be tractable. Similarly, the 1 added to the confidence level when adopting a trusted associate's opinion must remain constant for all vertices and trusted associates within a trusted component so that the expected time for X_s to leave a trusted component can be calculated. Distinguishing between trusted associates and non-trusted associates allows the process to reset itself in a Markov-like property. When treating all neighbors

the same, there is no control over how fast the confidence levels are increasing.

Our model of confidence levels presented here was chosen primarily for tractability. However, there are other possibilities both for how the confidence level is updated and for how the confidence level affects the time period until a vertex updates again. Perhaps a more natural choice for how the confidence level updates is that the confidence level increases by one if v_i adopts v_j's opinion and v_j's opinion is the same as v_i's. If v_i's and v_j's opinions are different, then v_i's confidence level decreases by 1. A different model would be to reset v_i's confidence level to 0 in the latter case. Confidence levels could also be bounded above and stay at the upper bound once reached until they decrease. The confidence level could also increase or decrease by amounts determined by which neighbor the opinion was adopted from.

The choice of exponential distributions for the time periods between updates is common in stochastic models. In our model they are crucial for being able to define the holding time at each vertex for the dual random walk X_s. However, if other techniques can be used, then other distributions could be considered. The use of the differences $f(c) - f(c-1)$ is purely for notational ease and does not restrict the waiting function f. The main restriction on f from the point of view of analysis of the model is that the limiting probability of the semi-Markov process must exist. One necessary condition is that the expected holding times of $\widetilde{X}_s$ are finite. However, it seems that little is known about the existence of the limiting probability when the expected holding times are infinite. This remains an interesting open question.

One reason our modification to the voter model seems appealing is that it is a stochastic system with nontrivial dependence of the limiting behavior on the structure of G but which still remains tractable. It would be very interesting to construct other modifications of the voter model that also have this property, or to construct similar modifications of other stochastic models such as the contact process or the exclusion process (see [20]). As mentioned in the introduction, confidence levels were introduced originally in some simplistic deterministic models of opinion formulation. It would be interesting to incorporate confidence levels into other deterministic models, such as the majority process or

the k-threshold process (see [8]).

Since we were motivated to consider the voter model by our interest in epidemiological models, it would be interesting to consider other modifications that are more biologically motivated. Confidence levels can be thought of as modeling an individual's resistance to infection, but perhaps there are other ways of modeling resistance. More complicating factors such as mutations and length of infection would also be interesting to consider.

References

[1] D. J. Aldous and J. A. Fill, *Reversible Markov Chains and Random Walks on Graphs*, monograph in preparation, available from `http://www.stat.berkeley.edu/users/aldous`.

[2] S. J. Brams and P. C. Fishburn, *Approval Voting*, Birkhauser, Boston, 1983.

[3] P. Clifford and A. Sudbury, A Model for Spatial Conflict, *Biometrika*, 60 (1973), 581-588.

[4] N. C. Dalkey, The Delphi Method: An Experimental Study of Group Opinion, Memorandum RM-5888-PR, RAND Corporation, Santa Monica, June 1969.

[5] M. H. DeGroot, Reaching a Consensus, *Journal of the American Statistical Association*, 69 (1974), 167-182.

[6] J.-P. Doignon and J.C. Falmagne, Well-graded Families of Relations, *Discrete Mathematics*, 173 (1997), 35-44.

[7] P. Donnelly and D. Welsh, Finite Particle Systems and Infection Models, *Math. Proc. Comb. Phil. Soc.*, 94 (1983), 167-182.

[8] P. Dreyer, Applications and Variations of Domination in Graphs, Ph.D. dissertation, Rutgers University, 2000.

[9] R. Durrett, *Lecture Notes on Particle Systems and Percolation*, Wadsworth and Brooks/Cole, Pacific Grove, California, 1988.

[10] J.-C. Falmagne, Stochastic Token Theory, *Journal of Mathematical Psychology*, 41 (1997), 129-143.

[11] J.-C. Falmagne and M. Regenwetter, A Random Utility Model for Approval Voting, *Journal of Mathematical Psychology*, 40 (1996), 152-159.

[12] J. R. P. French, A Formal Theory of Social Power, *Psychology Review*, 63 (1956), 181-194.

[13] R. G. Gallager, *Discrete Stochastic Processes*, Kluwer Academic Publishers, Boston, 1996, 174-178.

[14] D. Griffeath, *Additive and Cancellative Interacting Particle Systems*, Lecture Notes in Mathematics, vol. 724, Springer-Verlag, Berlin, 1979.

[15] F. Hoffman, presentation at the Twenty-Sixth Southeastern International Conference on Combinatorics, Graph Theory, and Computing. Held at Florida Atlantic University in Boca Raton, Florida, March 6-10, 1995.

[16] R. Holley and T. M. Liggett, Ergodic Theorems for Weakly Interacting Systems and the Voter Model, *Ann. Probab.*, 3 (1975), no. 4, 643-663.

[17] B. Latané, The Emergence of Clustering and Correlation from Social Interaction, in: R. Hegeselmann and H. O. Peitgen, eds., *Models of Social Dynamics: Order, Chaos, and Complexity*, Holder-Pichler-Tempsky, Vienna, 1996, 79-104.

[18] B. Latané and E. Fink, Symposium: Dynamical Social Impact Theory and Communication, *Journal of Communication*, 46, no. 4 (1996), 4-77.

[19] B. Latané and A. Nowak, Attitudes as Catastrophes: From Dimensions to Categories with Increasing Involvement, in: R. R. Vallacher and A. Nowak, eds., *Dynamical Systems in Social Psychology*, Academic Press, New York, 1994, 219-249.

[20] T. M. Liggett, *Interacting Particle Systems*, Grundlehren der mathematicshen Wissenschaften, no. 276, Springer-Verlag, New York, 1985.

[21] T. M. Liggett, *Stochastic Interacting Systems: Contact, Voter and Exclusion Processes*, Grundlehren der mathematicshen Wissenschaften, no. 324, Springer-Verlag, Berlin, 1999.

[22] H. A. Linstone and M. Turoff, eds., *The Delphi Method: Techniques and Applications*, Addison-Wesley, Reading, MA, 1975.

[23] L. Lovász, Random Walks on Graphs: A Survey, in: *Combinatorics, Paul Erdős is Eighty*, vol. 2, Bolyai Soc. Math. Stud., 2, János Bolyai Math. Soc., Budapest, 1996.

[24] S. Merrill, *A Unified Theory of Voting: Directional and Proximity Spatial Models*, Cambridge University Press, New York, 1999.

[25] S. Poljak and M. Sûra, On Periodical Behavior in Society with Symmetric Influences, *Combinatorica*, 3 (1983), 119-121.

[26] F. S. Roberts, *Discrete Mathematical Models, with Applications to Social, Biological, and Environmental Problems*, Prentice-Hall, Upper Saddle River, New Jersey, 1976, 258-363.

[27] F. S. Roberts, unpublished notes, 1999.

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 *S*usceptibles, *I*nfected, and *R*ecovered), 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.

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

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. 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 $A \subseteq D_k$ such that $|A| \geq 1 + \sum_{i=0}^{k-1}(a_i - f)$ 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 firefighters placed in D_{n+1} for times $1, \ldots, n$. 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 $v_{\rightarrow i}$ 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, $v_{\rightarrow i}$ 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 $v_{\rightarrow 1}$ 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 $v \in A^j_{r_j}$, add $v_{\rightarrow j}$ 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^i_r, 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 $A^{i'}_r$ is nonempty, form a set $\widehat{A^{i'}_r} \subseteq D^{d-1}_{k-r} \subseteq \mathbb{L}^{d-1}$ by eliminating the i' coordinate of each element in $A^{i'}_r$. By (3.6), $\left|N(\widehat{A^{i'}_r}) \cap D^{d-1}_{k-r+1}\right| \geq \left|\widehat{A^{i'}_r}\right| + d - 2$. 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 $v \in A_m^{i'}$, 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,

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

Since $\left| A_m^{i'} \right| \geq 1$, (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_r^i, 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 $A_r^{i'}$ is nonempty, or equivalently, the largest i' coordinate of elements of A, and let $r' \neq m$ be the minimum value of r where $A_r^{i'}$ is nonempty. If $\left| A_m^{i'} \right| \geq 2$, then using the same construction as in the $n_{i'} \geq 3$ case, we have by (3.7)

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

If $\left| A_m^{i'} \right| = 1$, then we again use the construction from the $n_{i'} \geq 3$ case. However, $\left| \widehat{A_{r'}^{i'}} \right| \geq 2d-3$, so by induction, $\left| N(\widehat{A_{r'}^{i'}}) \cap D_{k-r'+1}^{d-1} \right| \geq \left| \widehat{A_{r'}^{i'}} \right| + 2d - 4$. Here, the notation D_z^{d-1} means the set $D_z \subseteq \mathbb{L}^{d-1}$, emphasizing the dimension of $\mathbb{L}^{d-1}$. For each v in $N(\widehat{A_{r'}^{i'}}) \cap D_{k-r'+1}^{d-1}$,

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,

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

□

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 $(0, 0, \ldots, x_i, \ldots, 0)$, 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,

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

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. □

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 $|A| \geq \frac{3}{2}f^2$, 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 $|A| \geq \frac{3}{2}f^2$. 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 $|A| \geq \frac{3}{2}f^2$, 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 $\sum_{r=t}^{m} \sigma_r < \frac{3}{2} f^2$, 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 $\sigma_t + 1$, and by using (3.10), is equal to $\sigma_m + (m - t + 1)$. Since $g(\sigma) = \sigma_m + (m - t + 1) < f$ and $\sigma_m > 0$, we have that

$$\sigma_m < f \tag{3.11}$$

$$\text{and } m - t + 1 < f. \tag{3.12}$$

Thus,

$$\begin{aligned}\sum_{r=t}^{m} \sigma_r &= \sum_{r=t}^{m} (\sigma_m + (m - r)) \text{ by (3.10)} \\ &= \sigma_m(m - t + 1) + \frac{(m - t)(m - t + 1)}{2} \\ &< \frac{3}{2} f^2, \text{ by (3.11) and (3.12).}\end{aligned}$$

□

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 $\left|D_k^+\right| \geq \frac{3}{2} f^2$. 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,

$$\left|N(D_k^+) \cap D_{k+1}^+\right| \geq \left|D_k^+\right| + f \geq \frac{3}{2} f^2 + f,$$

and so

$$\left|N(r) \cap D'_1\right| \geq \left|D'_0\right| + \left(\frac{3}{2}f^2 - 1\right) + f.$$

If $A' \subseteq D'_i$, where $i > 0$ and $|A'| \geq \frac{3}{2}f^2$, 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 of D^+_k in $\mathbb{L}^3$. □

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 $P_n \times P_n \times P_n$. Then the maximum number of vertices which can be saved by deploying one firefighter per time step with an initial outbreak at v (for any choice of 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 will actually prove a stronger statement. Consider the graph G induced from the lattice $\mathbb{L}^3$ by vertices with non-negative coordinates and coordinate sum at most $3n$. We will prove the theorem for the graph G. Note that G contains $P_n \times P_n \times P_n$ as an induced subgraph.

Let r_t be the number of firefighters deployed at vertices with coordinate sum greater than t by time t, and let b_t be the number of vertices with coordinate sum exactly t which are burned at time t. We claim that $b_t - r_t \geq \frac{t^2+t+2}{2}$ for all t regardless of what firefighter placements are made. There are $\binom{t+2}{2} = \frac{t^2+3t+2}{2}$ vertices with coordinate sum exactly t, so this statement is saying that at time t, the number of reserve firefighters together with the number of vertices with coordinate sum t which are unburned cannot exceed t. Considering up to $t = 3n$, when all vertices have had a chance to be burned, we obtain that at most $1 + 2 + \ldots + 3n = O(n^2)$ vertices are unburned. This implies the same statement for the grid graph.

The proof of the claim is by induction. At time 0, there are no reserve firefighters, and one vertex with coordinate sum 0 is burned; their 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. There are at least $\frac{t^2+t+2}{2} > \frac{t(t+1)}{2}$ burned vertices at time t, so by Lemma 3.7, at time $t+1$ there are at least $t + 2$ more potentially burned vertices. So if no firefighters prevent spread at this time, then we have $r_{t+1} = r_t + 1$ (the extra firefighter being deployed), and $b_{t+1} \geq b_t + (t + 2)$, so $b_{t+1} - r_{t+1} \geq b_t - r_t + (t+1) \geq \frac{t^2+t+2}{2} + (t+1) = \frac{t^2+3t+4}{2} = \frac{(t+1)^2+(t+1)+2}{2}$, proving the claim. However, using a reserve firefighter to reduce the spread of the fire decreases r_{t+1} and b_{t+1} both by one from their would-be values, which does not change their difference. Consequently, regardless of the deployment of the firefighters at time $t + 1$, the claim is satisfied, which completes the proof of the theorem. □

In practice, one can ensure (with an outbreak at $(0, 0, 0)$) that t vertices with coordinate sum 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" at distance k from (n, n, n), consisting of all of the vertices that (grid) distance from this antipodal vertex; 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 $P_n \times P_n \times P_n$ 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

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

$$\begin{aligned}
\text{subject to:} & b_{x,t} + d_{x,t} - b_{y,t-1} \geq 0, \text{ for all } x \in L,\ y \in N(x), \text{ and } 1 \leq t \leq T, && (3.13)\\
& b_{x,t} + d_{x,t} \leq 1, \text{ for all } x \in L \text{ and } 1 \leq t \leq T, && (3.14)\\
& b_{x,t} - b_{x,t-1} \geq 0, \text{ for all } x \in L \text{ and } 1 \leq t \leq T, && (3.15)\\
& d_{x,t} - d_{x,t-1} \geq 0, \text{ for all } x \in L \text{ and } 1 \leq t \leq T, && (3.16)\\
& \sum_{x \in L} (d_{x,t} - d_{x,t-1}) \leq 2, \text{ for } 1 \leq t \leq T, && (3.17)\\
& b_{x,0} = \begin{cases} 1 & \text{if } x \text{ is the origin,} \\ 0 & \text{otherwise,} \end{cases} \text{ for all } x \in L, && (3.18)\\
& d_{x,0} = 0, \text{ for all } x \in L, && (3.19)\\
& b_{x,t}, d_{x,t} \in \{0, 1\}, \text{ for all } x \in L \text{ and } 0 \leq t \leq T. && (3.20)
\end{aligned}$$

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

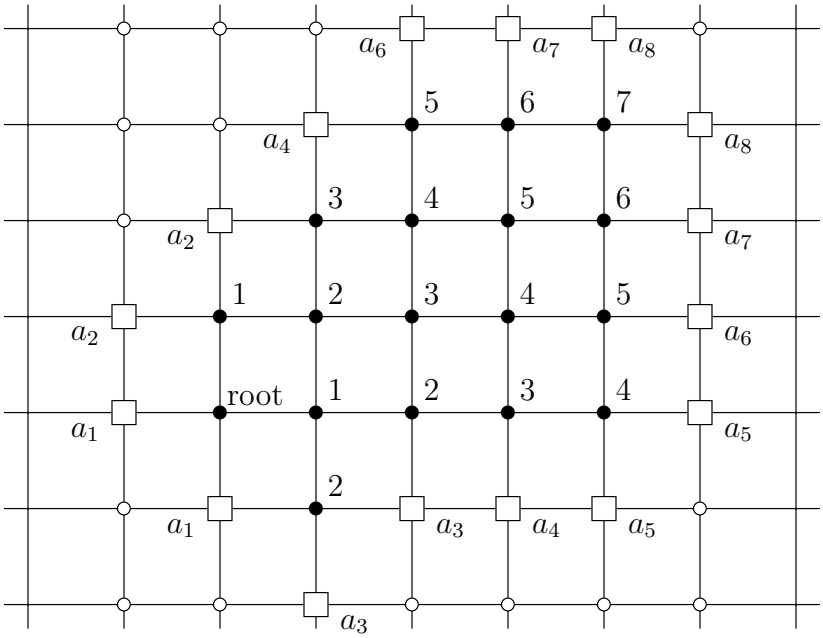

Figure 3.1: Optimal solution of the integer program used in the proof of Theorem 3.11. The fire outbreak starts at time 0 at the root, and then spreads to the black vertices at the times written next to the vertices. The square firefighters a_i are placed at time i. This placement of two firefighters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

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) *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.

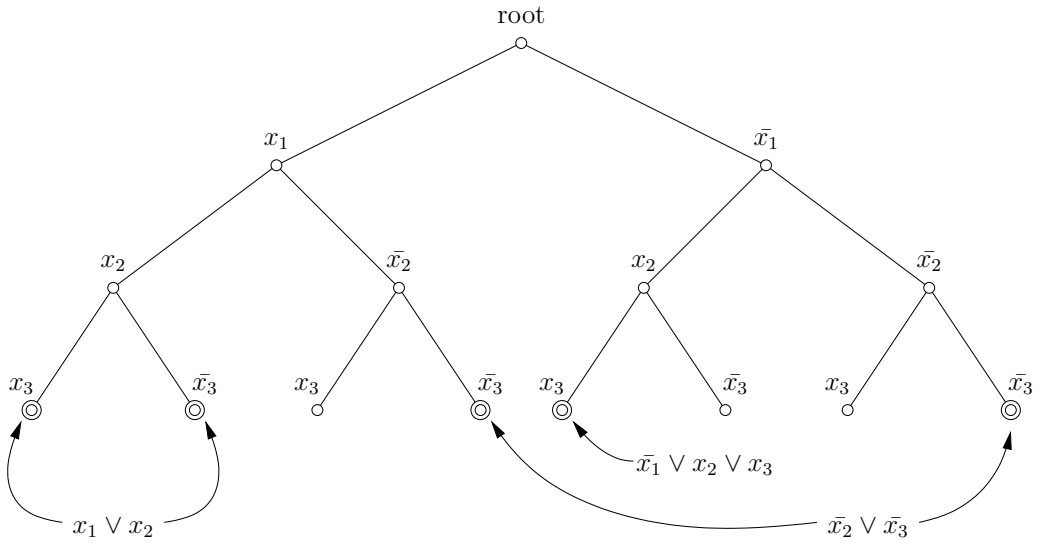

Figure 3.2: Reduction of the formula $\varphi = (x_1 \vee x_2) \wedge (\bar{x_1} \vee x_2 \vee x_3) \wedge (\bar{x_2} \vee \bar{x_3})$ to a binary tree.

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

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

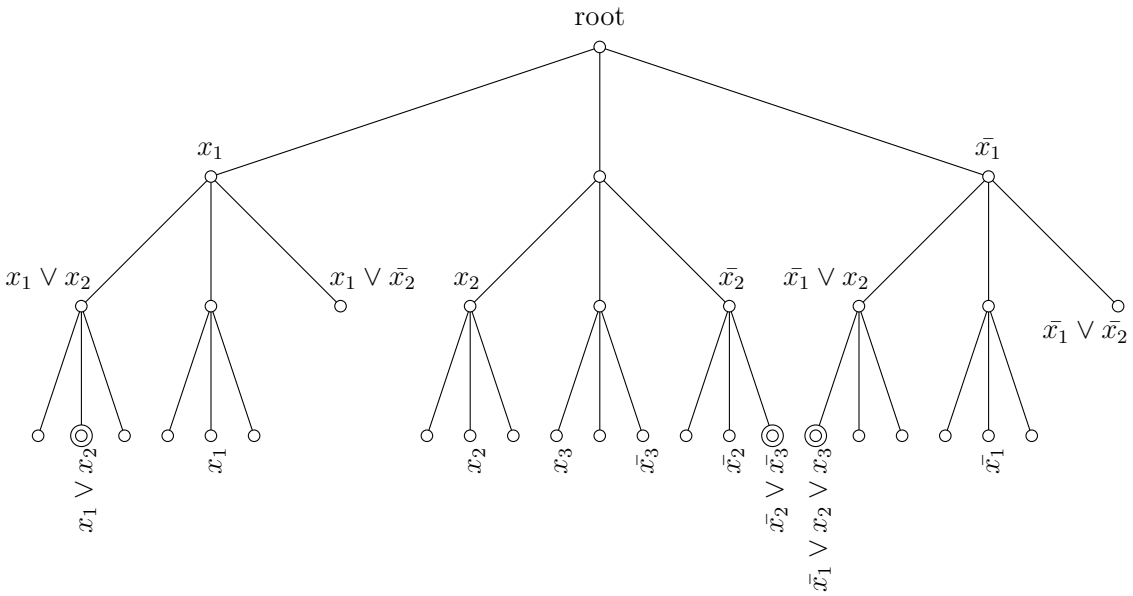

Figure 3.3: Construction of the tree T_1 for $\varphi = (x_1 \vee x_2) \wedge (\bar{x_1} \vee x_2 \vee x_3) \wedge (\bar{x_2} \vee \bar{x_3})$. Clause vertices of φ are marked with double circles, and clause vertices of $\widetilde{\varphi}$ are labeled. Vertices on level 1 that contain x_1 or $\bar{x_1}$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

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 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

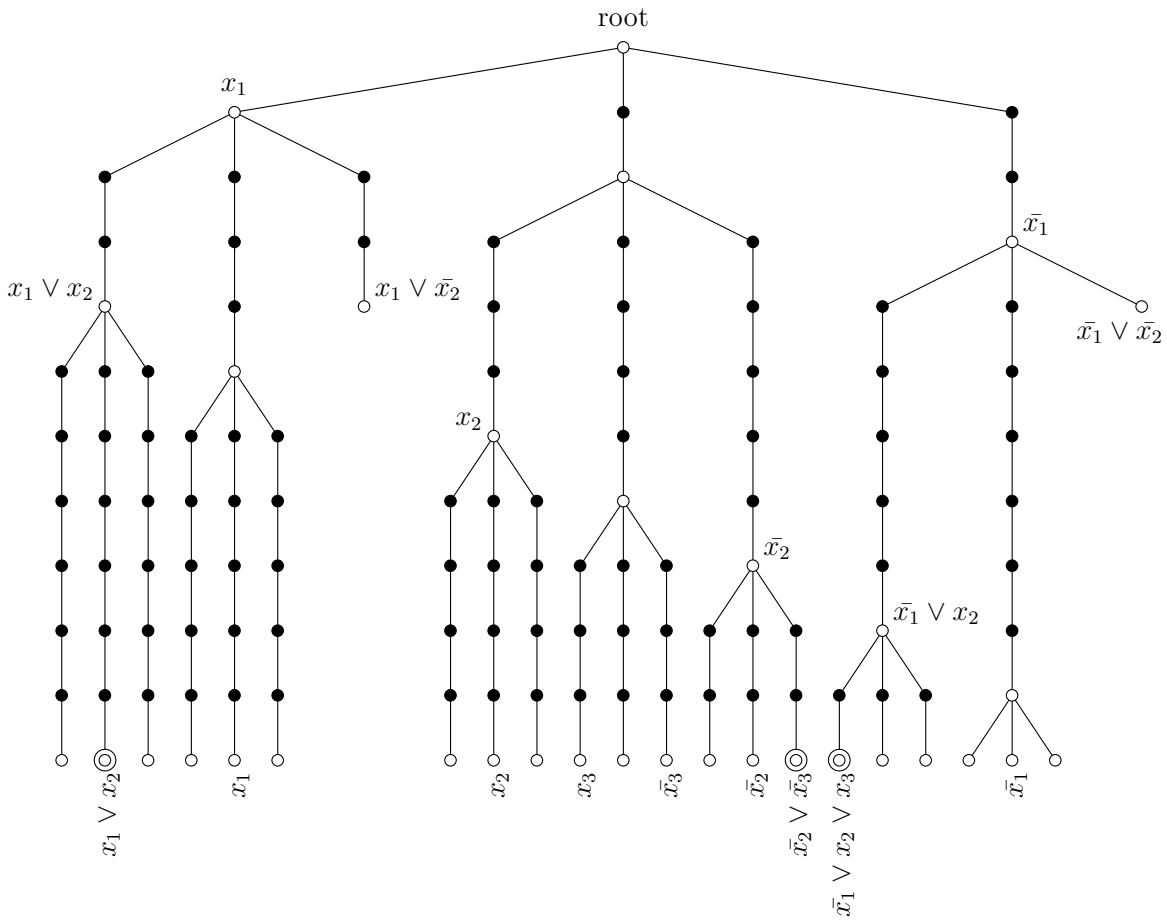

Figure 3.4: Construction of the tree T_2 for $\varphi = (x_1 \vee x_2) \wedge (\bar{x_1} \vee x_2 \vee x_3) \wedge (\bar{x_2} \vee \bar{x_3})$. Vertices that are also in T_1 are marked with hollow dots, and new vertices are marked with black dots.

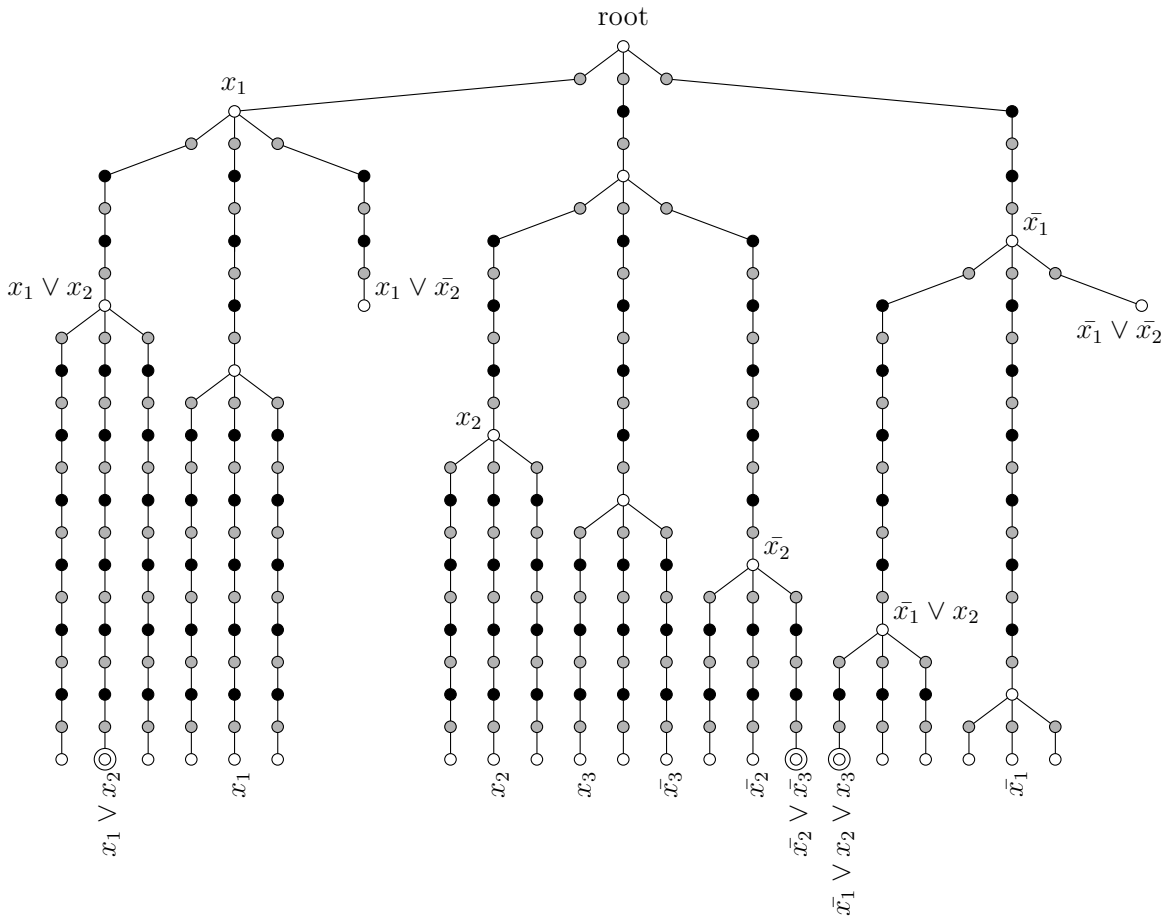

Figure 3.5: Construction of the tree T_3 for $\varphi = (x_1 \vee x_2) \wedge (\bar{x_1} \vee x_2 \vee x_3) \wedge (\bar{x_2} \vee \bar{x_3})$. Vertices that are also in T_1 are marked with hollow dots, and new vertices on subdivided edges are marked with gray dots.

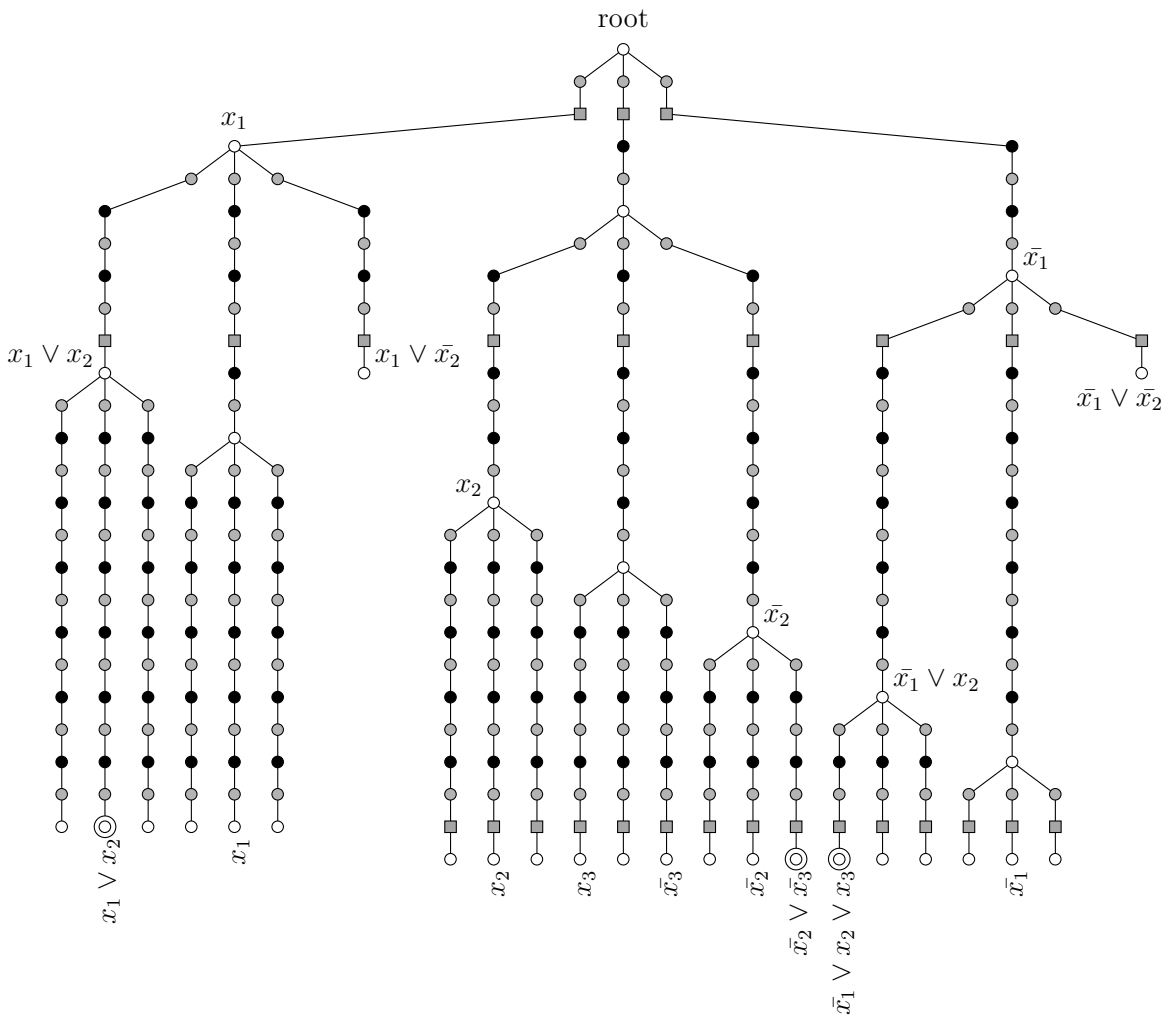

Figure 3.6: Construction of the tree T_4 for $\varphi = (x_1 \vee x_2) \wedge (\bar{x_1} \vee x_2 \vee x_3) \wedge (\bar{x_2} \vee \bar{x_3})$. Vertices that are also in T_1 are marked with hollow dots, vertices added when forming T_3 are marked by gray dots, and those added when forming T_4 are marked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

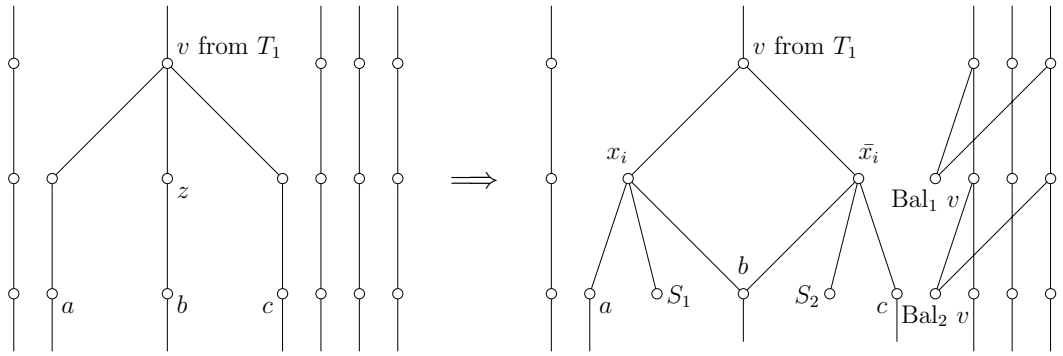

Figure 3.7: The decision widget for x_i. The vertices v, a, b, and c shown in T_4 on the left correspond to the vertices with the same labels shown in G_5 on the right. The vertex labeled z is removed when forming G_5.

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, 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 $\text{Bal}_1\ v$ and $\text{Bal}_2\ v$, where $\text{Bal}_1\ v$ is on the level below v and $\text{Bal}_2\ v$ is two levels below. Connect $\text{Bal}_1\ v$ 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 $\text{Bal}_2\ v$ 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 $\text{Bal}_1\ v$ 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 $\text{Bal}_1\ v$ and $\text{Bal}_2\ v$ 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

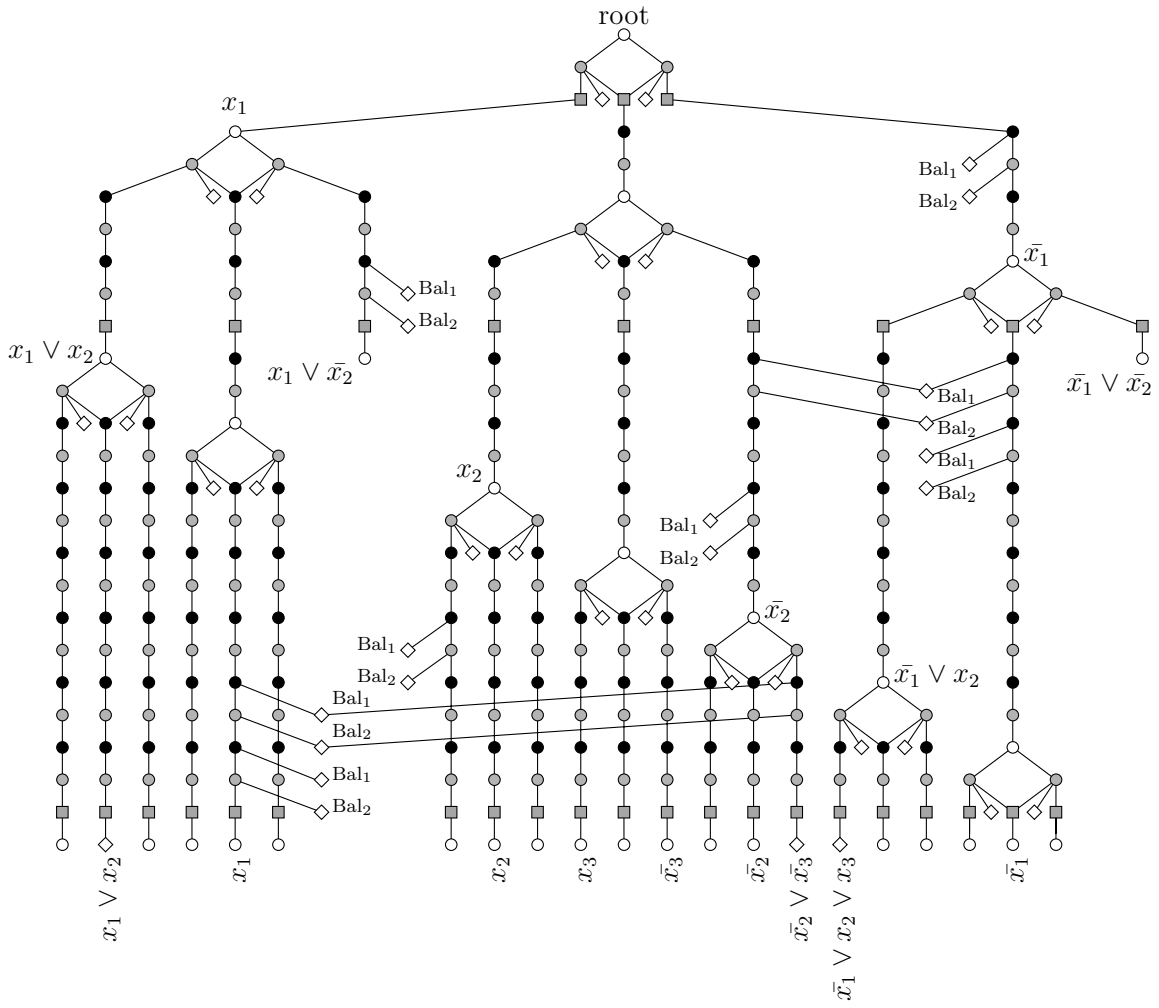

Figure 3.8: Construction of the graph G_5 for $\varphi = (x_1 \vee x_2) \wedge (\bar{x_1} \vee x_2 \vee x_3) \wedge (\bar{x_2} \vee \bar{x_3})$. Vertices added when forming T_3 are marked by gray dots, and those added when forming T_4 are marked by gray squares. Super-spreader vertices are marked with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

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 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

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

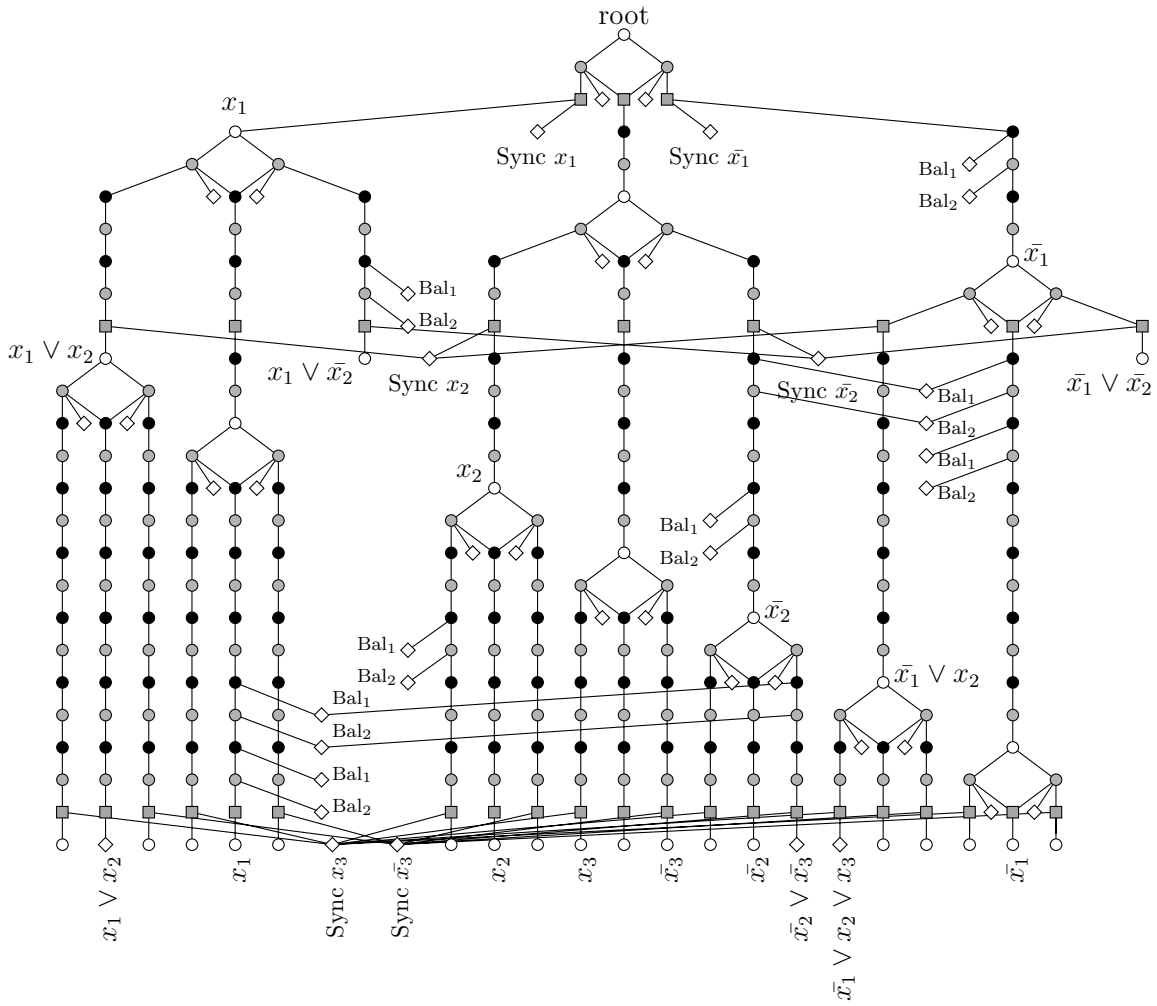

Figure 3.9: Construction of the graph G_6 for $\varphi = (x_1 \vee x_2) \wedge (\bar{x_1} \vee x_2 \vee x_3) \wedge (\bar{x_2} \vee \bar{x_3})$. Vertices added when forming T_3 are marked by gray dots, and those added when forming T_4 are marked by gray squares. Super-spreader vertices are marked with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

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

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

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, Bal_1 v on level j and 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 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 Bal_1 v on level j, and the one firefighter available at time $j + 1$ must be used to defend 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 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 $j \leq j_i$ 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 $a_3 = \text{Sync } x_i$. If x_1 is false in τ, then set $a_1 = \bar{x_1}$, $a_2 = S_1$, and $a_3 = \text{Sync } \bar{x_i}$. 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 Bal_1 v on level j is saved. Similarly, either the vertices S_1 and S_2 on level $j+1$ are saved or 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 Bal_1 v is not saved, then choose a_j to be 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 Bal_2 v is not saved, then choose a_{j+1} to be 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.

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 $b_i \to g_j$. To determine the chargings, compare $\mathrm{wt}(b_i)$ to $\mathrm{wt}(g_i)$. If $\mathrm{wt}(b_i) \leq \mathrm{wt}(g_i)$, then the greedy algorithm is doing well compared to the optimal, and we charge b_i to g_i. If $\mathrm{wt}(b_i) > \mathrm{wt}(g_i)$, 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 of vertices saved by this optimal sequence to that of the greedy algorithm by using the standard combinatorial technique of counting in two

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

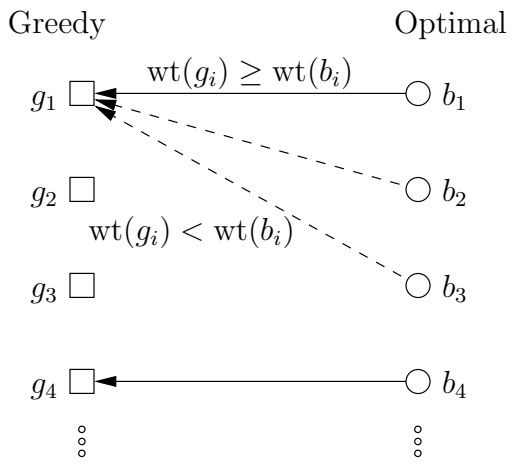

Figure 3.10: Pictorially "charging" b_i to g_j.

different ways:

$$\begin{aligned}\#\text{ of vertices saved by optimal sequence} &= \sum_{i=1}^{k} \mathrm{wt}(b_i) \\ &= \sum_{j=1}^{\ell} \left(\sum_{i:b_i \to g_j} \mathrm{wt}(b_i) \right).\end{aligned}$$

To bound $\sum_{i:b_i \to g_j} \mathrm{wt}(b_i)$, 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,

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

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

$$\begin{aligned}\#\text{ of vertices saved by optimal sequence} &= \sum_{j=1}^{\ell} \left(\sum_{i:b_i \to g_j} \mathrm{wt}(b_i) \right) \\ &\leq \sum_{j=1}^{\ell} 2\mathrm{wt}(g_j) \\ &= 2(\#\text{ of vertices saved by the greedy algorithm}).\end{aligned}$$

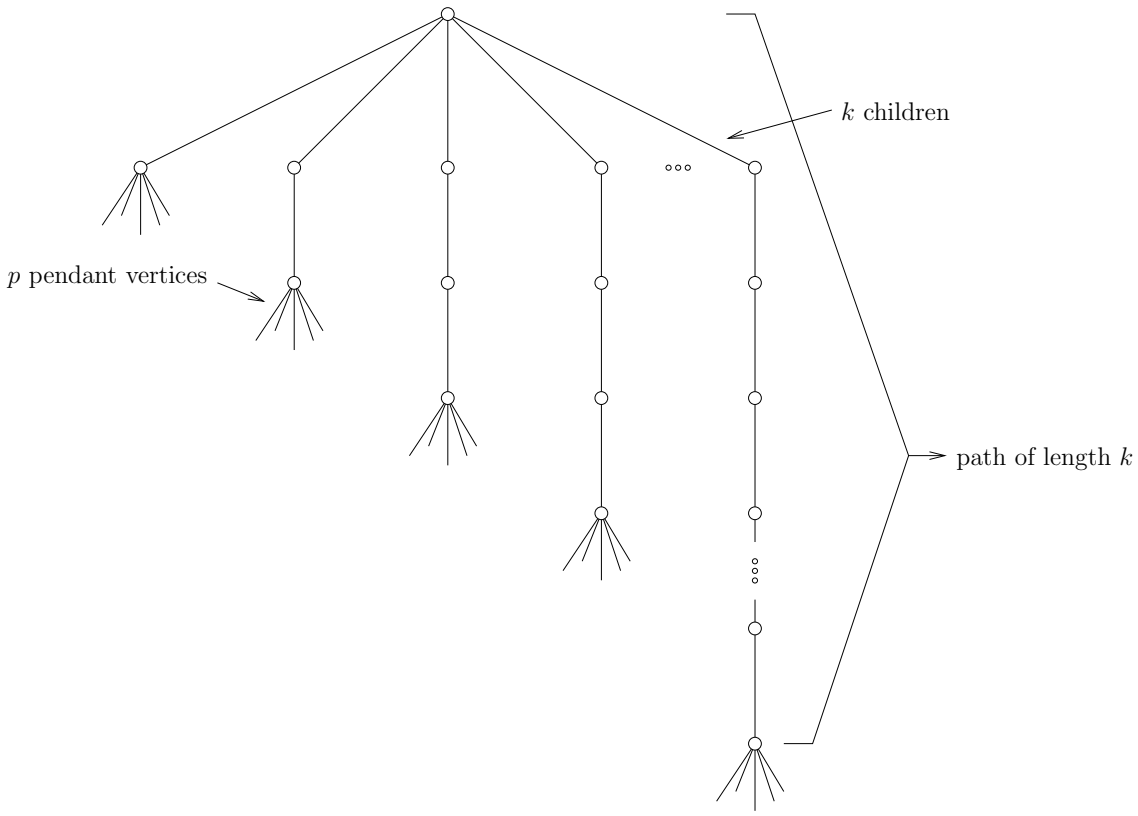

Figure 3.11: Construction showing that asymptotically the greedy algorithm saves 1/2 of the number of vertices saved by an optimal firefighter sequence.

Hence, the greedy algorithm saves at least half as many vertices as 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 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 $v = u$.

$$\begin{aligned} \text{maximize} \sum_{v \in V(G)} & \operatorname{wt}(v)x(v) \\ \text{subject to:} \sum_{v \text{ on level } \ell} & x(v) \leq 1, \text{ for each level } \ell, && (3.22) \\ \sum_{v \succeq u} & x(v) \leq 1, \text{ for each leaf } u, && (3.23) \\ & x(v) \in \{0,1\}, \text{ for each vertex } v. && (3.24) \end{aligned}$$

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 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

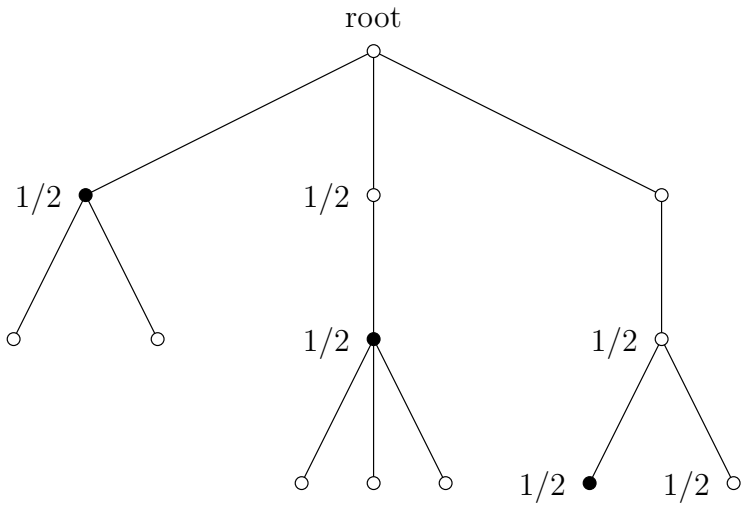

Figure 3.12: In this example on 13 vertices, the LP optimal is 8.5, whereas the IP optimal is 8. The nonzero values of $x(v)$ for the LP optimal solution appear next to the vertices, and the optimal firefighter sequence is indicated with black vertices.

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 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

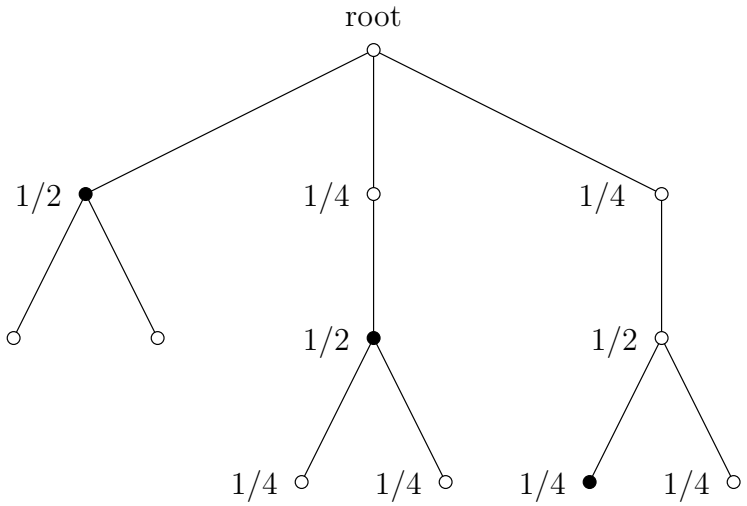

Figure 3.13: In this example on 12 vertices, the LP optimal when using constraint (3.26) is 7.5, whereas the IP optimal is 7. The nonzero values of $x(v)$ for the LP optimal solution appear next to the vertices, and the optimal vaccination sequence is indicated with black vertices.

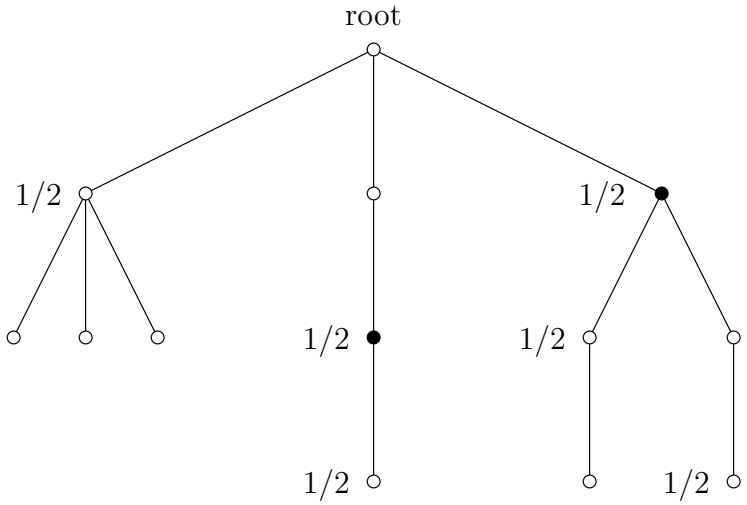

Figure 3.14: In this example on 13 vertices, the LP optimal when using constraint (3.27) is 7.5, whereas the IP optimal is 7. The nonzero values of $x(v)$ for the LP optimal solution appear next to the vertices, and the optimal vaccination sequence is indicated with black vertices.

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.*

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 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

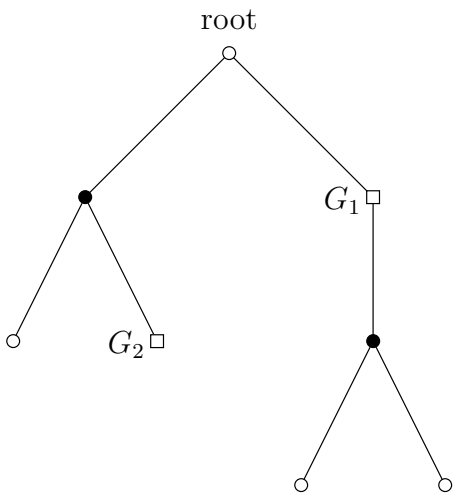

Figure 3.15: When defending one child per burnt vertex in this tree, the greedy algorithm is not optimal. The greedy algorithm defends the square vertices G_1 and G_2, saving 5 vertices, while the optimal firefighter sequence marked with black dots saves 6 vertices.

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.

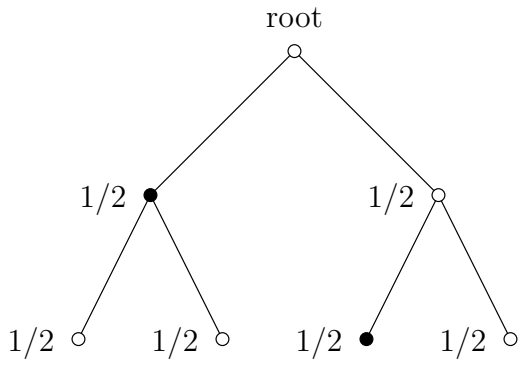

Figure 3.16: This example demonstrates the model of one defended child per burnt vertex. In this example on 7 vertices, the LP optimal is 5, whereas the IP optimal is 4. The nonzero values of $x(v)$ for the LP optimal solution appear next to the vertices, and the optimal firefighter sequence is indicated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

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.

Chapter 4

The Elimination Procedures for the Competition Number and the Phylogeny Number

4.1 Introduction

Given an acyclic digraph D, the competition graph $C(D)$ is defined to be the undirected graph with $V(D)$ as its vertex set and where vertices x and y are adjacent if there exists another vertex z such that the arcs (x, z) and (y, z) are both present in D. Competition graphs were introduced by Cohen [1] to study ecosystems. The vertices of an acyclic digraph D, known as a food web, represent species, and the arc (x, z) indicates that z is a prey of x. An edge exists between two vertices x and y in $C(D)$ if and only if x and y have a common prey. In addition to ecology, competition graphs have also found application in studying communication over noisy channels, interfering radio transmissions, and models of complex economic and energy problems – see the discussions in Raychaudhuri and Roberts [9] and Roberts [12]. Lundgren [6], Roberts [10], and Kim [3] survey the extensive literature of competition graphs.

In [11] Roberts noted that for any graph G, G along with r isolated vertices is the competition graph of some acyclic digraph if r is sufficiently large. The competition number $k(G)$ is defined to be the least such r. In general, determining the competition number of a graph is difficult: Opsut [7] showed that this problem is NP-complete. Kim and Roberts in [11] and [4] have determined the competition number of graphs with 0, 1, and 2 triangles, but for few other graph classes is the competition number known. As another approach, Roberts considered using an elimination procedure to calculate $k(G)$. An elimination procedure takes as input G and an ordering $\mathcal{O} = v_1, \ldots, v_n$ of the vertices of G and produces an acyclic digraph D such that $C(D) = G \cup I_r$; that is, the competition

graph of D is G along with r isolated vertices. The procedure "eliminates" each vertex in order by ensuring that all of the edges incident on the vertex will appear in $C(D)$. The goal is to create an elimination procedure that for some ordering $\mathcal{O}$ outputs an acyclic digraph D where $|V(D) \setminus V(G)| = k(G)$.

Elimination procedures which seek to determine a graph-theoretical parameter through step-wise elimination of vertices have various applications in graph theory. A common example are algorithms for determining if a graph is chordal by finding perfect elimination orders such that each vertex is "simplicial" in the graph of remaining vertices. Roberts [11] was led to consider an elimination procedure for the competition number through variants of perfect elimination used by Parter [8], Rose [15], and Golumbic [2] in connection with numerical analysis. Here elimination procedures are used to find a good order for eliminating variables during Gaussian elimination of a matrix.

Opsut [7] found an example of a graph G where Roberts' original elimination procedure does not calculate the competition number $k(G)$, thus giving a counterexample to Roberts' conjecture that the procedure always calculates $k(G)$. Kim and Roberts [5] then modified the elimination procedure and asked whether their modified procedure works for all graphs. They were able to show that the modified version calculates the competition number for a large class of graphs, the so-called "kite-free" graphs.

In this chapter, we present a new, simpler proof of Kim and Roberts' theorem that their elimination procedure calculates the competition number for kite-free graphs. We also present a graph L where Kim and Roberts' elimination procedure does not always calculate the competition number, in the following sense: for each order $\mathcal{O}$ of vertices of L, the elimination procedure can produce an acyclic digraph with more than $k(L)$ additional vertices.

Phylogeny graphs are related to competition graphs, and were introduced by Roberts and Sheng [13] from an idealized model for reconstructing phylogenetic trees. Given an acyclic digraph D, the phylogeny graph $P(D)$ is defined to be the undirected graph with $V(D)$ as its vertex set and with adjacencies as follows: two vertices x and y are adjacent if one of the arcs (x, y) or (y, x) is present in D, or if there exists another vertex z such

that the arcs (x, z) and (y, z) are both present in D. Roberts and Sheng noted that for any simple graph G, G is an induced subgraph of $P(D)$ for some acyclic digraph D. The phylogeny number $p(G)$ is the least number r such that D has $|V(G)| + r$ vertices. Determining the phylogeny number of a graph was shown by Roberts and Sheng [13] to be NP-complete, and they also determined the phylogeny number for connected graphs with 0, 1, and 2 triangles [14].

Phylogenetic tree reconstruction deals with establishing evolutionary relationships between different species. A phylogenetic tree is a rooted directed tree, where the species are vertices and an arc (x, z) indicates that z is a direct ancestor of x. Given a set of species and a measure of similarity between each pair of species, we wish to create a phylogenetic tree where similar species are closely related. The concept of "closely related" can be defined in many ways, and Roberts and Sheng choose the tree metric where the distance between x and y is the shortest distance to a common ancestor. If the similarity measure is purely "similar" or "not similar," then we can encode the similarity relationship in a graph G, where two vertices are adjacent if and only if the corresponding species are similar. If we relax the condition of finding a tree, then finding a phylogenetic acyclic digraph D for the set of species turns out to be the same as finding an acyclic digraph D such that $P(D) = G$. The number of assumptions made in defining phylogeny graphs considerably removes the concept from the original biological motivation. However, phylogeny graphs give a starting point for studying phylogenetic tree reconstruction, and the related competition graphs give rise to many interesting mathematical techniques and questions for phylogeny graphs. One question that we address in section 4.5 is the construction and analysis of an elimination procedure. For a survey on competition graphs and phylogeny graphs and their relation, see Roberts [10].

Both of the results mentioned above for competition numbers carry over to phylogeny numbers: we present an analogous elimination procedure for the phylogeny number, show that it calculates the phylogeny number for kite-free graphs, and show that the procedure does not calculate the phylogeny number for the graph L. The phylogeny case is simpler than the competition case because of how vertices are "handled" when eliminated, and

allows greater insight into the concepts underlying these results.

We mention here that there are many other variants of the competition number besides the phylogeny number. The common enemy graph of a digraph $D = (V, A)$ is the graph with vertex set V and where vertices x and y are adjacent if and only if there is a vertex a in D such that (a, x) and (a, y) are arcs of D (a is a common enemy or predator of x and y). In the niche graph of D, x and y are adjacent if and only if there is a vertex a such that (a, x) and (a, y) are arcs of D *or* there is a vertex b such that (x, b) and (y, b) are arcs of D (x and y *either* have a common predator a *or* a common prey b). In the competition-common enemy graph of D, x and y are adjacent if and only if there is a vertex a such that (a, x) and (a, y) are arcs of D *and* there is a vertex b such that (x, b) and (y, b) are arcs of D (x and y have *both* a common prey *and* a common predator). The common enemy number, niche number, and competition-common enemy number of a graph G can be defined analogously to competition number, and we refer the reader to [10] for a survey of results about these graph parameters. No one has considered elimination procedures for these parameters yet, and it would be interesting to see what results about the elimination procedures for the competition number and the phylogeny number carry over.

Note that the focus of creating elimination procedures is not on efficiency, since calculating the competition number or the phylogeny number with an elimination procedure requires $n!$ runs (one for each ordering of the vertices). As mentioned above, calculating both the competition number and the phylogeny number have been shown to be NP-complete. Instead, the focus is on whether an elimination procedure *could* be created that exactly calculates the relevant number. This is interesting both for historical reasons and because many of the practical examples are relatively small, exactness is sometimes more important than efficiency. Our results provide a better understanding of why Kim and Roberts' procedure is exact for kite-free graphs, and the counterexample suggests that creating an elimination procedure that is exact for all graphs might be much more difficult than originally thought.

In this chapter, the graph G that we wish to calculate the competition or phylogeny

number of need not be connected. For convenience, we will sometimes also describe a subgraph H of a graph G only as "consisting of" certain edges of G. It is understood that H has no isolated vertices: the vertices of H are only the endpoints of edges in H.

4.2 The Elimination Procedure for the Competition Number

We will first formalize our definitions and describe Kim and Roberts' elimination procedure using our terminology; however, its workings are the same as the elimination procedure described in [5].

Definition 4.1. Let $D = (V, A)$ be an acyclic digraph. The *competition graph* $C(D)$ is a simple graph with vertex set V where two vertices x and y are adjacent in $C(D)$ if there exists a vertex z such that both (x, z) and (y, z) are arcs in D. From the ecological origins of competition graphs, z is known as a *prey* of x if (x, z) is an arc of D.

Definition 4.2. For a simple graph G, the *competition number* $k(G)$ is the least number r such that there exists an acyclic digraph D on $|V(G)| + r$ vertices where $C(D)$ is G along with r isolated vertices.

Before presenting the formal description of the elimination procedure for the competition number, we first give an informal description. Given a graph G and an ordering $\mathcal{O} = v_1, \ldots, v_n$ of the vertices of G, we eliminate each vertex iteratively, in the process building up an acyclic digraph D with the desired properties. When eliminating vertex v_i, we "cover" every edge incident to v_i that has not been covered in a previous iteration. By "covering" an edge e, we mean that the appropriate arcs and possibly vertices have been added to D so that e is an edge in $C(D)$. The subgraph G_i is a spanning subgraph of G that contains the edges of G that have not been covered in an iteration prior to the i^{th} iteration. The subgraph G_i' consists of the edges of G_i that are incident on v_i, and so the edges of G_i' must be covered in the i^{th} iteration. Cliques are used to maximize the coverage of G_i' using the least number of added vertices. If C is a clique, then by adding arcs in D_i from the vertices of C to a common vertex x, all of the edges in C appear in $C(D_i)$. Thus, all of the vertices in C are "preying" on the same "species" x, and hence

competing with each other.

The improvement of Kim and Roberts' modified elimination procedure over Roberts' original procedure was in recognizing that the edges in G'_i are the *only* edges that must be covered in the i^{th} iteration. For choosing the cliques they utilize the subgraph H_i consisting of the edges from v_i to vertices of higher index. The cliques covering G'_i are chosen from H_i, even though some of the edges in H_i might already be covered. By using maximal cliques of H_i, possibly more uncovered edges that are not in G'_i will be covered.

Definition 4.3. Let $E_G(v)$ denote the subgraph of G with vertex set $N_G[v]$ and containing only those edges of G incident to the vertex v.

The Kim-Roberts Elimination Procedure for the Competition Number[1]

Input: A graph G and an ordering $\mathcal{O} = v_1, v_2, \ldots, v_n$ of the vertices of G.

Output: An acyclic digraph $D := D_n$ such that $C(D)$ is G with some additional isolated vertices.

Initialization: Set D_0 to the digraph with vertex set $V(G)$ and no arcs. D_i is an acyclic digraph constructed at the i^{th} iteration.

Set $G_1 := G$. G_i is a spanning subgraph of G that contains the edges of G that do not appear in $C(D_{i-1})$.

Set $S_1 := \emptyset$. S_i is a set of vertices available as prey.

i^{th} Iteration, $i = 1, \ldots, n$: Set G'_i to $E_{G_i}(v_i)$, and set H_i to the subgraph of G induced by $\{v_i\} \cup \{v_j : j > i \text{ and } v_j \in N_G(v_i)\}$. Let $\mathcal{E}_i = \{C_1, \ldots, C_k\}$ be a minimum size edge covering of G'_i by maximal cliques of H_i, ordered arbitrarily. Form G_{i+1} from G_i by removing the edges of C_j from G_i for all j.

Form the digraph D_i by adding vertices and arcs to D_{i-1} as follows: Pick k distinct vertices $u_1, \ldots, u_k$ from S_i. If $|S_i| < k$, then add $k - |S_i|$ additional vertices

[1] If we refer simply to "the elimination procedure for the competition number," then we mean the Kim-Roberts elimination procedure.

$u_{k-|S_i|}, \ldots, u_k$ to D_i. For each clique $C_j \in \mathcal{E}_i$, add the arcs (w, u_j) to D_i for each $w \in C_j$. Form S_{i+1} by $S_{i+1} := (S_i \setminus \{u_1, \ldots, u_k\}) \cup \{v_i\}$.

Remark 4.4. Note that finding a minimum size edge covering of G'_i by maximal cliques of H_i is equivalent to finding a minimum size vertex covering by maximal cliques of $H_i \setminus \{v_i\}$ of the subgraph induced by $N_{G_i}(v_i)$. The transformation between these procedures is as follows: For each clique C_j in $\mathcal{E}_i = \{C_1, \ldots, C_k\}$, set $\widehat{C}_j = C_j \setminus \{v_i\}$. Then $\widehat{\mathcal{E}}_i = \{\widehat{C}_1, \ldots, \widehat{C}_k\}$ is a minimum size vertex cover of $N_{G_i}(v_i)$ by maximal cliques of $H_i \setminus \{v_i\}$ if and only if $\mathcal{E}_i$ is a minimum size edge cover of G'_i by maximal cliques of H_i.

To help analyze the workings of the elimination procedure, we now introduce a more generalized elimination procedure. In the generalized elimination procedure, a clique cover of the entire graph G is given, and from this covering and the order of the vertices we construct D.

The Generalized Elimination Procedure for the Competition Number

Input: A graph G, an ordering $\mathcal{O} = v_1, v_2, \ldots, v_n$ of the vertices of G, and an edge clique covering $\mathcal{G}$ of G.

Output: An acyclic digraph $D := D_n$ such that $C(D)$ is G with some additional isolated vertices.

Initialization: Set D_0 to the digraph with vertices $V(G)$ and no arcs. D_i is an acyclic digraph constructed at the i^{th} iteration.

Set $S_1 := \emptyset$. S_i is a set of vertices available as prey.

i^{th} Iteration, $i = 1, \ldots, n$: Let $\mathcal{G}_i = \{C_1, \ldots, C_k\}$ be the subset of $\mathcal{G}$ where for each $C_j \in \mathcal{G}_i$, v_i is the vertex in C_j of least index. Order $\mathcal{G}_i$ arbitrarily.

Form the digraph D_i by adding vertices and arcs to D_{i-1} as follows: Pick k distinct vertices $u_1, \ldots, u_k$ from S_i. If $|S_i| < k$, then add $k - |S_i|$ additional vertices $u_{k-|S_i|}, \ldots, u_k$ to D_i. For each clique $C_j \in \mathcal{G}_i$, add the arcs (w, u_j) to D_i for each $w \in C_j$. Form S_{i+1} by $S_{i+1} := (S_i \setminus \{u_1, \ldots, u_k\}) \cup \{v_i\}$.

The following proposition is Proposition 1 from [5], if we note that though the proof is worded only for the elimination procedure, it also applies to the generalized elimination procedure.

Proposition 4.5. *The generalized elimination procedure for the competition number produces an acyclic digraph D where $C(D)$ is G along with some additional isolated vertices.*

We now show that the Kim-Roberts elimination procedure is a special case of the generalized elimination procedure.

Lemma 4.6. *Let $\mathcal{E}_i$ be the edge coverings generated by the Kim-Roberts elimination procedure for a graph G and a vertex ordering $\mathcal{O} = v_1, \ldots, v_n$. Then the set $\mathcal{E} = \bigcup_{i=1}^{n} \mathcal{E}_i$ is an edge clique covering of G.*

Proof. Since each $\mathcal{E}_i$ is chosen to be a set of cliques of H_i and H_i is a subgraph of G, $\mathcal{E}$ is a set of cliques of G. We now show that $\mathcal{E}$ covers all the edges of G. Let $\{v_k, v_\ell\}$ be an edge in G, where $k < \ell$. Suppose that $\{v_k, v_\ell\}$ is not an edge in any clique of $\bigcup_{i=1}^{k-1} \mathcal{E}_i$. Then G_k contains the edge $\{v_k, v_\ell\}$, as does G'_k. Since $\mathcal{E}_k$ is an edge clique covering of G'_k, there will exist a clique $C_j \in \mathcal{E}_k$ that contains $\{v_k, v_\ell\}$. Therefore, $\mathcal{E} = \bigcup_{i=1}^{n} \mathcal{E}_i$ is an edge clique covering of G. □

Proposition 4.7. *Let G be a graph and $\mathcal{O} = v_1, \ldots, v_n$ be an ordering of the vertices of G. Then the number of vertices added to the digraph produced by the Kim-Roberts elimination procedure is the number of vertices added to the digraph produced by the generalized elimination procedure if the edge clique covering $\mathcal{G}$ is chosen to be $\mathcal{E}$ as defined in Lemma 4.6.*

Proof. The proposition follows from Lemma 4.6 and the observation that the subsets $\mathcal{G}_i$ used in the generalized elimination procedure are exactly the subsets $\mathcal{E}_i$ used in the elimination procedure. □

Furthermore, if in each i^{th} iteration the same clique C_1 is chosen from $\mathcal{G}_i = \mathcal{E}_i$, then the two digraphs are isomorphic.

In order to analyze the number of additional vertices needed by the elimination procedure to construct D, we would like a formula expressing this number in terms of the cliques chosen. We will give such a formula and show its correctness via the generalized elimination procedure.

Definition 4.8. The *elimination number* $M(G, \mathcal{O}, \mathcal{G})$ of a graph G, an ordering $\mathcal{O}$ of the vertices, and an edge clique covering $\mathcal{G}$ is the number of vertices added to G so that $C(D)$ is $G \cup I_{M(G,\mathcal{O},\mathcal{G})}$, where D is the digraph produced by the generalized elimination procedure for the competition number with G, $\mathcal{O}$, and $\mathcal{G}$ as inputs.

Definition 4.9. Let G be a graph, $\mathcal{O} = v_1, v_2, \ldots, v_n$ be an ordering of the vertices of G, and $\mathcal{G} = \{C_1, C_2, \ldots, C_k\}$ be an edge clique covering of G. For each vertex v_i, let $\mathcal{G}_i$ be the subset of $\mathcal{G}$ where for each $C_j \in \mathcal{G}_i$, v_i is the vertex in C_j of least index. Recursively define the integer sequences $\{a_i^{\mathcal{G}}\}_{i=0}^n$ and $\{b_i^{\mathcal{G}}\}_{i=1}^n$ by

$$\begin{aligned} a_0^{\mathcal{G}} &= 0, \\ b_i^{\mathcal{G}} &= \max\{|\mathcal{G}_i| - a_{i-1}^{\mathcal{G}}, 0\}, \\ a_i^{\mathcal{G}} &= a_{i-1}^{\mathcal{G}} - (|\mathcal{G}_i| - b_i^{\mathcal{G}}) + 1. \end{aligned}$$

Define

$$h_G(\mathcal{G}, \mathcal{O}) = \sum_{i=1}^{n} b_i^{\mathcal{G}}.$$

Note that $a_i^{\mathcal{G}}$ is the number $|S_i|$ of available prey at the end of the i^{th} iteration, and $b_i^{\mathcal{G}}$ is the number of new vertices added to D_i in the i^{th} iteration.

Lemma 4.10. *Let G be a graph, $\mathcal{O} = v_1, v_2, \ldots, v_n$ be an ordering of the vertices of G, and $\mathcal{G} = \{C_1, C_2, \ldots, C_k\}$ be an edge clique covering of G. Then $M(G, \mathcal{O}, \mathcal{G}) = h_G(\mathcal{G}, \mathcal{O})$.*

Proof. Note that $\mathcal{G}_i$ is defined in exactly the same way in both the generalized elimination procedure and in Definition 4.9. Summing $b_i^{\mathcal{G}}$ over all iterations, we get

$$M(G, \mathcal{O}, \mathcal{G}) = |V(D) \setminus V(G)| = \sum_{i=1}^{n} b_i^{\mathcal{G}} = h_G(\mathcal{G}, \mathcal{O}). \qquad \square$$

By taking a minimum over all edge clique covers and vertex orders of G, we can use $h_G(\mathcal{G}, \mathcal{O})$ to calculate the competition number of G.

Lemma 4.11. *For a graph G, the competition number $k(G)$ equals $\min_{\mathcal{G}} \min_{\mathcal{O}} h_G(\mathcal{G}, \mathcal{O})$, where $\mathcal{G}$ ranges over all edge clique coverings of G and $\mathcal{O}$ ranges over all orderings of the vertices of G.*

Proof. Let $\mathcal{G} = \{C_1, C_2, \ldots, C_k\}$ be an edge clique covering of G and $\mathcal{O} = v_1, v_2, \ldots, v_n$ be an ordering of the vertices of G. By Proposition 4.5, the generalized elimination procedure produces an acyclic digraph D such that $C(D)$ is G with some additional isolated vertices. By Lemma 4.10, $M(G, \mathcal{O}, \mathcal{G}) = h_G(\mathcal{G}, \mathcal{O})$, and so $k(G) \leq \min_{\mathcal{G}} \min_{\mathcal{O}} h_G(\mathcal{G}, \mathcal{O})$.

Now let F be an acyclic digraph that attains the competition number for G; that is, $C(F)$ is G along with isolated vertices and $|V(F) \setminus V(G)| = k(G)$. Let $\mathcal{O} = v_1, v_2, \ldots, v_n$ be an ordering of the vertices of G such that if (v_ℓ, v_k) is an arc in F, then $k < \ell$. Such an ordering exists because F is acyclic. We construct an edge clique covering $\mathcal{G}$ of G from F as follows: For a vertex $v_i \in V(G)$, $N_F^{\text{in}}(v_i)$ induces a clique in G, and for a vertex $b \in V(F) \setminus V(G)$, $N_F^{\text{in}}(b)$ induces a clique in G. Since having arcs into a common prey is the only way edges can be present in G, $\mathcal{G}$ is an edge clique cover of G. Now observe that the digraph D produced by the generalized elimination procedure with $\mathcal{G}$ and $\mathcal{O}$ has the same number of vertices as F. In fact, if the appropriate $u_1, \ldots, u_k$ are chosen from S_i, then D is isomorphic to F.

Therefore, $k(G) = |V(F) \setminus V(G)| = |V(D) \setminus V(G)| = M(G, \mathcal{O}, \mathcal{G}) \geq \min_{\mathcal{G}} \min_{\mathcal{O}} h_G(\mathcal{G}, \mathcal{O})$, and so $k(G) = \min_{\mathcal{G}} \min_{\mathcal{O}} h_G(\mathcal{G}, \mathcal{O})$. □

We now specialize the definition of the elimination number.

Definition 4.12. Given a graph G and an ordering $\mathcal{O}$, let $\mathcal{E} = \mathcal{E}(G, \mathcal{O}) = \{\mathcal{E}_1, \mathcal{E}_2, \ldots, \mathcal{E}_n\}$ be edge clique coverings obtained during the Kim-Roberts elimination procedure. Of course, the notation is ambiguous since the way to choose the $\mathcal{E}_i$ is not completely specified in the procedure. The *elimination number* $M(G)$ is the minimum of $M(G, \mathcal{O}, \mathcal{E})$ over all orders $\mathcal{O}$ and some $\mathcal{E}$ obtained when using $\mathcal{O}$. Kim and Roberts show that for certain classes of graphs, if $M(G)$ is this minimum and is attained for $\mathcal{O}$ and some $\mathcal{E}$, then it is attained for the same $\mathcal{O}$ and any $\mathcal{E}$ corresponding to $\mathcal{O}$. If this is the case, $M(G)$ is unambiguously defined.

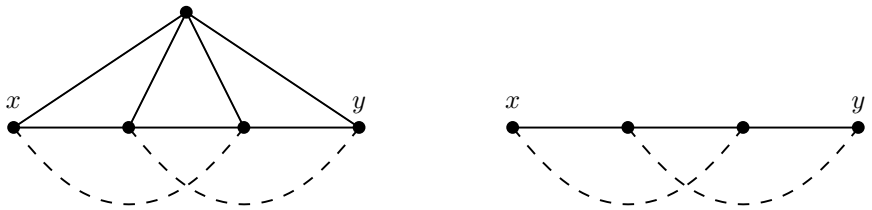

Figure 4.1: A kite and a kite-body.

The determination of necessary and sufficient conditions for $M(G)$ to be unambiguously defined is an interesting open problem. Lemma 4.11 shows that there is always a "right" clique cover for each order such that the minimum over orders attains the competition number $k(G)$. Kim and Roberts show that there is a class of graphs, known as the kite-free graphs, for which $M(G)$ is unambiguously defined and equals $k(G)$. In the next section, we present a new and simpler proof of this result. Kim and Roberts also asked if $M(G)$ is unambiguously defined and equals $k(G)$ for all graphs. However, in section 4.4 we exhibit a graph L such that for each order $\mathcal{O}$ there is a choice of clique cover $\mathcal{E}_i$ in the Kim-Roberts elimination procedure such that $M(G, \mathcal{O}, \mathcal{E}) > k(G)$. This answers the Kim-Roberts question negatively.

4.3 Kite-free Graphs

In this section, we present a new and simpler proof of Kim and Roberts' theorem in [5] that their elimination procedure for competition numbers is exact for kite-free graphs.

Definition 4.13. A *kite* is the left configuration shown in Figure 4.1. In a kite, the solid edges must be present, and the dotted edges cannot be present. The edge between vertices x and y may or may not be present. A *kite-free* graph does not have a kite as a configuration, meaning that neither of the two graphs on five vertices that are kites are present as induced subgraphs. A *kite-body* is the right configuration shown in Figure 4.1. Again, the solid edges must be present, the dotted edges cannot be present, and the edge between x and y may or may not be present. Similarly, a *kite-body-free* graph does not have a kite-body as a configuration.

The following lemma is Lemma 3 from [5].

Lemma 4.14. *Let G be a kite-body-free graph, S a subset of $V(G)$, H an induced subgraph of G, and $C_1, C_2, \ldots, C_k$ a vertex cover of S using maximal cliques of H. If a subset T of S forms a clique in H, then T is contained in some C_ℓ.*

Lemma 4.15. *Let G be a kite-free graph and $\mathcal{O} = v_1, \ldots, v_n$ be an ordering of the vertices of G. In the Kim-Roberts elimination procedure, an edge $\{v_j, v_k\}$ with $v_j, v_k \in N_G(v_i)$ appears in some clique of $\mathcal{E}_\ell$, where $\ell \leq i$.*

Proof. Suppose that $\{v_j, v_k\}$ with $v_j, v_k \in N_G(v_i)$ does not appear in any clique of $\mathcal{E}_\ell$, where $\ell < i$. Observe that in the elimination procedure all edges incident on a vertex v_ℓ are covered by $\bigcup_{r=1}^{\ell} \mathcal{E}_r$. Thus, $k > i$ and $j > i$, and so $\{v_j, v_k\}$ is an edge in $H_i \setminus \{v_i\}$. We now consider three different cases. Suppose that both v_j and v_k are in $N_{G_i}(v_i)$. Let $\mathcal{E}_i = \{C_1, \ldots, C_s\}$, and set $\widehat{C}_t = C_t \setminus \{v_i\}$. As stated in Remark 4.4, $\widehat{\mathcal{E}}_i = \{\widehat{C}_1, \ldots, \widehat{C}_s\}$ is a vertex cover of $N_{G_i}(v_i)$ by maximal cliques of $H_i \setminus \{v_i\}$. Since v_j and v_k are in $N_{G_i}(v_i)$, the edge $\{v_j, v_k\}$ forms a clique in $N_{G_i}(v_i)$. Since G is kite-free, $H_i \setminus \{v_i\}$ is kite-body-free. By Lemma 4.14, $\{v_j, v_k\}$ is a clique contained in some $\widehat{C}_\ell$, and so appears in clique C_ℓ of $\mathcal{E}_i$.

For the second case, suppose that neither v_j nor v_k is in $N_{G_i}(v_i)$. Since v_j is not in $N_{G_i}(v_i)$, the edge $\{v_j, v_i\}$ appears in some clique of $\mathcal{E}_\ell$, where $\ell < i$. Since v_k is also not in $N_{G_i}(v_i)$, then $\{v_k, v_i\}$ appears in some clique of $\mathcal{E}_m$, where $m < i$. If $\ell = m$, then by the first case applied to v_ℓ, $\{v_j, v_k\}$ appears in a clique of $\mathcal{E}_\ell = \mathcal{E}_m$. Thus, $\ell \neq m$. Since $\{v_j, v_k\}$ is not in any clique of $\mathcal{E}_\ell$ or $\mathcal{E}_m$, the edges $\{v_j, v_m\}$ and $\{v_k, v_\ell\}$ do not appear in G. But then the vertices $v_i, v_j, v_k, v_\ell, v_m$ form a kite in G, contradicting the kite-free-ness of G.

We now consider the third case, where, without loss of generality, $v_j \notin N_{G_i}(v_i)$ and $v_k \in N_{G_i}(v_i)$. As in the second case, the edge $\{v_j, v_i\}$ appears in some clique of $\mathcal{E}_\ell$ where $\ell < i$. Since $v_k \in N_{G_i}(v_i)$, there exists a clique C of $\mathcal{E}_i$ that contains the edge $\{v_k, v_i\}$. If C does not contain $\{v_j, v_k\}$, there must exist a vertex v_p in C such that v_p is not adjacent to v_j. Otherwise, C could be expanded to include v_j, contradicting the fact that C is

a maximal clique of H_i. Thus, the edges $\{v_j, v_p\}$ and $\{v_k, v_\ell\}$ do not appear in G. But then the vertices $v_i, v_j, v_k, v_\ell, v_p$ form a kite in G, again contradicting the kite-free-ness of G. □

Definition 4.16. Let G be a graph, and $\mathcal{O} = v_1, \ldots, v_n$ be an ordering of the vertices of G. For each $i = 1, \ldots, n$, define

$$T_i = \{v_j : j > i,\ v_j \text{ is adjacent to } v_i, \text{ and}$$
$$\nexists\ k < i \text{ where } v_k \text{ is adjacent to both } v_i \text{ and } v_j\}.$$

Define $\widehat{G}_i$ to be the subgraph of G with vertices $T_i \cup \{v_i\}$ and edges $\{\{x, v_i\} : x \in T_i\}$.

Lemma 4.17. *Let G be a graph, $\mathcal{O} = v_1, \ldots, v_n$ be an ordering of the vertices of G, and $\mathcal{G}$ be an edge clique covering of G. Then the cliques of $\mathcal{G}_i$ must cover $\widehat{G}_i$.*

Proof. Let $e = \{v_i, v_j\}$ be an edge of $\widehat{G}_i$ and C a clique of $\mathcal{G}$ that covers e. Note that $i < j$. Since v_i is an endpoint of e, the least index of a vertex in C is at most i. Suppose the vertex of least index in C is v_k, where $k < i$. But then v_k is adjacent to both v_i and v_j, contradicting the construction of $\widehat{G}_i$. Thus, v_i is the vertex of least index in C, and hence C is in $\mathcal{G}_i$. Therefore, $\widehat{G}_i$ is covered by cliques of $\mathcal{G}_i$. □

Lemma 4.18. *Let G be a kite-free graph and $\mathcal{O} = v_1, \ldots, v_n$ be an ordering of the vertices of G. Then the subgraphs G'_i of G generated by the Kim-Roberts elimination procedure are exactly $\widehat{G}_i$.*

Proof. Observe that only edges whose endvertices are adjacent to v_i can appear in cliques of $\mathcal{E}_i$. Let $\{v_j, v_i\}$ be an edge where $j > i$ and where there exists a $k < i$ such that v_k is adjacent to both v_i and v_j. By Lemma 4.15, $\{v_j, v_i\}$ appears in some clique of $\mathcal{E}_\ell$, $\ell \leq k$, and so $\{v_j, v_i\}$ is not in G'_i. If there is no such $k < i$, then no $\mathcal{E}_\ell$ can cover the edge $\{v_j, v_i\}$ where $\ell < i$, and by Lemma 4.15, $\{v_j, v_i\}$ is then covered by a clique of $\mathcal{E}_i$. Thus, the definition of $\widehat{G}_i$ precisely describes G'_i. □

Lemma 4.19. *Let G be a kite-free graph, $\mathcal{O} = v_1, \ldots, v_n$ be an ordering of the vertices of G. Then $h_G(\mathcal{E}, \mathcal{O}) = \min_{\mathcal{G}} h_G(\mathcal{G}, \mathcal{O})$, where $\mathcal{E}$ is any edge clique cover produced by the*

Kim-Roberts elimination procedure for the competition number on G and $\mathcal{O}$, and where the right-hand side minimum is taken over all edge clique covers $\mathcal{G}$ of G.

Proof. Let $\mathcal{G}$ be an edge clique cover of G such that $h_G(\mathcal{G}, \mathcal{O})$ is minimized. By Lemma 4.17, $\mathcal{G}_i$ must cover $\widehat{G}_i$ for all i. But by Lemma 4.18, $G'_i = \widehat{G}_i$. Since $\mathcal{E}_i$ is chosen to be a minimum size cover of G'_i, $|\mathcal{E}_i| \leq |\mathcal{G}_i|$.

We now show that $a_i^{\mathcal{E}} \geq a_i^{\mathcal{G}}$ and $b_i^{\mathcal{E}} \leq b_i^{\mathcal{G}}$ for all i. Suppose, for contradiction, that there exists an i such that our desired conditions fail. Let i be the least such index. Now,

$$\begin{aligned} b_i^{\mathcal{E}} &= \max\{|\mathcal{E}_i| - a_{i-1}^{\mathcal{E}}, 0\} \\ &\leq \max\{|\mathcal{G}_i| - a_{i-1}^{\mathcal{E}}, 0\} \quad \text{since } |\mathcal{E}_i| \leq |\mathcal{G}_i| \\ &\leq \max\{|\mathcal{G}_i| - a_{i-1}^{\mathcal{G}}, 0\} \quad \text{since } a_{i-1}^{\mathcal{E}} \geq a_{i-1}^{\mathcal{G}} \text{ by assumption} \\ &= b_i^{\mathcal{G}}. \end{aligned}$$

But

$$\begin{aligned} a_i^{\mathcal{E}} &= a_{i-1}^{\mathcal{E}} - (|\mathcal{E}_i| - b_i^{\mathcal{E}}) + 1 \\ &\geq a_{i-1}^{\mathcal{E}} - (|\mathcal{G}_i| - b_i^{\mathcal{E}}) + 1 \quad \text{since } |\mathcal{E}_i| \leq |\mathcal{G}_i| \\ &\geq a_{i-1}^{\mathcal{G}} - (|\mathcal{G}_i| - b_i^{\mathcal{E}}) + 1 \quad \text{since } a_{i-1}^{\mathcal{E}} \geq a_{i-1}^{\mathcal{G}} \text{ by assumption} \\ &\geq a_{i-1}^{\mathcal{G}} - (|\mathcal{G}_i| - b_i^{\mathcal{G}}) + 1 \quad \text{from above} \\ &= a_i^{\mathcal{G}}. \end{aligned}$$

Thus $a_i^{\mathcal{E}} \geq a_i^{\mathcal{G}}$ and $b_i^{\mathcal{E}} \leq b_i^{\mathcal{G}}$ for all i. Summing over i gives $h_G(\mathcal{E}, \mathcal{O}) \leq h_G(\mathcal{G}, \mathcal{O})$. □

Theorem 4.20. *[Kim and Roberts] For a kite-free graph G, the elimination number $M(G)$ is unambiguously defined and equals the competition number $k(G)$.*

Proof. By Lemma 4.11, $k(G) = \min_{\mathcal{G}} \min_{\mathcal{O}} h_G(\mathcal{G}, \mathcal{O})$, where $\mathcal{G}$ ranges over all edge clique coverings of G and $\mathcal{O}$ ranges over all orderings of the vertices of G. By Lemma 4.10, $M(G, \mathcal{O}, \mathcal{E}) = h_G(\mathcal{E}, \mathcal{O})$ for any $\mathcal{E}$ corresponding to $\mathcal{O}$. By Lemma 4.19, $h_G(\mathcal{E}, \mathcal{O}) = h_G(\mathcal{E}', \mathcal{O})$ for any $\mathcal{E}$ and $\mathcal{E}'$ produced by the Kim-Roberts elimination procedure and corresponding to $\mathcal{O}$. It follows that $M(G, \mathcal{O}, \mathcal{E})$ and therefore the elimination number $M(G)$ is unambiguously defined. Moreover, by Lemma 4.19, $M(G) = \min_{\mathcal{O}} M(G, \mathcal{O}, \mathcal{E}) = \min_{\mathcal{O}} \min_{\mathcal{G}} h_G(\mathcal{G}, \mathcal{O})$, and therefore, $M(G) = k(G)$. □

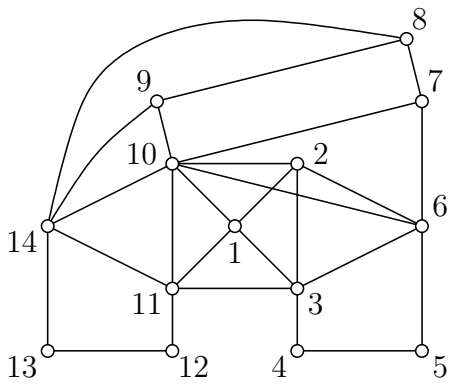

Figure 4.2: The graph L.

4.4 Counterexample Showing the Kim-Roberts Elimination Procedure Does Not Always Obtain $k(G)$

Theorem 4.20 states that if G is kite-free then $M(G)$ is unambiguously defined and equals $k(G)$. Kim and Roberts asked if this is true for all graphs. To answer this question negatively, we need to demonstrate a graph G such that for each order $\mathcal{O}$ there is a choice $\mathcal{E}$ of clique covers so that $M(G, \mathcal{O}, \mathcal{E}) > k(G)$. The graph L in Figure 4.2 is such a graph G.

From Theorem 4.20, any graph with the desired property must contain a kite. The graph L in Figure 4.2 contains two kites on the vertices $\{1, 2, 3, 10, 11\}$ and $\{1, 2, 3, 6, 10\}$. When eliminating vertices 1 or 2 first, two different clique covers of two triangles each can be used to eliminate the incident edges. One of these choices is a *good* choice for the edge clique cover, but one is a *bad* choice. Our effort in constructing the counterexample is to force 1 or 2 to be eliminated first, so that a bad choice can be made. When 1 or 2 is not eliminated first, then we will show that no choices allow the elimination procedure to attain the competition number.

Proposition 4.21. *For each ordering $\mathcal{O}$ of the vertices of the graph L in Figure 4.2, there is a choice of edge clique coverings $\mathcal{E}$ such that $M(L, \mathcal{O}, \mathcal{E}) > 2$.*

Proof. Let $\mathcal{O} = v_1, v_2, \ldots, v_{14}$ be an ordering of the vertices of L. We consider several cases:

Case 1. $v_1 = 1$.

We make the bad choice of the cliques $\{1, 2, 3\}$ and $\{1, 10, 11\}$. Any choice for v_2 other than vertex 2 can not be eliminated without increasing the number of extra vertices added to D since its remaining incident edge cannot be covered by a single clique. Thus, v_2 must be vertex 2. But after vertex 2 is eliminated, no vertex has its remaining incident edges coverable by a single clique. Thus, $M(L, \mathcal{O}, \mathcal{E}) > 2$ if $v_1 = 1$.

Case 2. $v_1 = 2$.

We make the bad choice of the cliques $\{2, 1, 3\}$ and $\{2, 6, 10\}$. Analogously to Case 1, vertex 1 is the only vertex that can then be eliminated as v_2 without increasing the number of added vertices, but after that no vertex has its remaining incident edges coverable by a single clique.

Case 3. $v_1 = 3$, 6, 10, 11, or 14.

Each of these vertices requires at least three cliques to cover its incident edges.

Case 4. $v_1 = 4$ or 5.

One of these vertices can be eliminated using two cliques, and the other is then the only vertex that can be eliminated without increasing the number of added vertices. But then no vertex has its remaining incident edges coverable by a single clique.

Case 5. $v_1 = 7$.

Vertex 7 can be eliminated with two cliques. Then vertices 8 and 9 in that order are the only vertices that can then be eliminated without increasing the number of added vertices. But after that no vertex has its remaining incident edges coverable by a single clique.

Case 6. $v_1 = 8$ or 9.

One of these vertices can be eliminated using two cliques, and then the other vertex and vertex 7 are the only vertices that can then be eliminated without increasing the number of added vertices. Vertex 7 must be eliminated after vertex 8 for this

to be the case. But after that no vertex has its remaining incident edges coverable by a single clique.

Case 7. $v_1 = 12$ or 13.

One of these vertices can be eliminated using two cliques, and the other is the only vertex that can then be eliminated without increasing the number of added vertices. But after that no vertex has its remaining incident edges coverable by a single clique.

Thus, there exists a choice $\mathcal{E}$ of clique cover such that $M(L, \mathcal{O}, \mathcal{E}) > 2$ for any order $\mathcal{O}$. □

Proposition 4.22. *The competition number of the graph L in Figure 4.2 is* 2.

Proof. First note that there is no vertex in L whose incident edges can be covered with one clique. Thus, $k(L) \geq 2$. But the elimination procedure using the order $1, 2, \ldots, 14$ and the good choice of cliques $\{1, 2, 10\}$ and $\{1, 3, 11\}$ for vertex 1 produces an elimination number $M(L, \mathcal{O}, \mathcal{E})$ of 2. Thus, $k(L) = 2$. □

4.5 The Elimination Procedure for the Phylogeny Number

The competition number problem is essentially a problem about minimum edge clique covers, where the "value" of a cover is computed in a weighted manner. The phylogeny number problem is similar in this regard. Thus, we can formulate an elimination procedure for the phylogeny number similar to that of the competition number and obtain analogous results.

Definition 4.23. Let $D = (V, A)$ be an acyclic digraph. The *phylogeny graph* $P(D)$ is a simple undirected graph with vertex set V and with adjacencies as follows: two vertices x and y are adjacent if one of the arcs (x, y) or (y, x) is present in D, or if there exists another vertex z such that the arcs (x, z) and (y, z) are both present in D.

Definition 4.24. For a simple graph G, the phylogeny number $p(G)$ is the least number r such that there exists an acyclic digraph D on $|V(G)| + r$ vertices where G is an induced subgraph of $P(D)$.

We now give the elimination procedure for the phylogeny number. Note that the only difference from the elimination procedure for the competition number is how edges of G are "accounted for" in D.

The Elimination Procedure for the Phylogeny Number

Input: A graph G and an ordering $\mathcal{O} = v_1, v_2, \ldots, v_n$ of the vertices of G.

Output: An acyclic digraph $D := D_n$ such that G is an induced subgraph of $P(D)$.

Initialization: Set D_0 to the digraph with vertex set $V(G)$ and no arcs. D_i is an acyclic digraph constructed at the i^{th} iteration.

Set $G_1 := G$. G_i is a spanning subgraph of G that contains the edges of G that do not appear in $P(D_{i-1})$.

i^{th} Iteration, $i = 1, \ldots, n$: Set G_i' to $E_{G_i}(v_i)$, and set H_i to the subgraph of G induced by $\{v_i\} \cup \{v_j : j > i \text{ and } v_j \in N_G(v_i)\}$. Let $\mathcal{E}_i = \{C_1, \ldots, C_k\}$ be a minimum size edge covering of G_i' by maximal cliques of H_i, ordered arbitrarily. Form G_{i+1} from G_i by removing the edges of C_j from G_i for all j.

Form the digraph D_i by adding vertices and arcs to D_{i-1} as follows: Add the arcs (w, v_i) to D_i for all vertices $w \in C_1 \setminus \{v_i\}$. For each clique $C_j \in \mathcal{E}_i \setminus \{C_1\}$, add a vertex b_j to $V(D_i)$ and add the arcs (w, b_j) to D_i for each $w \in C_j$.

We also give a generalized elimination procedure for the phylogeny number.

The Generalized Elimination Procedure for the Phylogeny Number

Input: A graph G, an ordering $\mathcal{O} = v_1, v_2, \ldots, v_n$ of the vertices of G, and an edge clique covering $\mathcal{G}$ of G.

Output: An acyclic digraph $D := D_n$ such that G is an induced subgraph of $P(D)$.

Initialization: Set D_0 to the digraph with vertices $V(G)$ and no arcs. D_i is an acyclic digraph constructed at the i^{th} iteration.

i^{th} Iteration, $i = 1, \ldots, n$: Let $\mathcal{G}_i = \{C_1, \ldots, C_k\}$ be the subset of $\mathcal{G}$ where for each $C_j \in \mathcal{G}_i$, v_i is the vertex in C_j of least index. Order $\mathcal{G}_i$ arbitrarily.

Form the digraph D_i by adding vertices and arcs to D_{i-1} as follows: Add the arcs (w, v_i) to D_i for all vertices $w \in C_1 \setminus \{v_i\}$. For each clique $C_j \in \mathcal{G}_i \setminus \{C_1\}$, add a vertex b_j to $V(D_i)$ and add the arcs (w, b_j) to D_i for each $w \in C_j$.

We will first show that the generalized elimination procedure produces an acyclic digraph, and then show that for the digraph D produced by the generalized elimination procedure, $P(D)$ has G as an induced subgraph.

Lemma 4.25. *Let D be the digraph produced by the generalized elimination procedure for the phylogeny number for a graph G, a vertex ordering $\mathcal{O} = v_1, \ldots, v_n$, and an edge clique covering $\mathcal{G}$. Then all vertices in $V(D) \setminus V(G)$ are sinks, and if (v_ℓ, v_k) is an arc, then $k < \ell$. Thus, D is acyclic.*

Proof. If $b \in V(D) \setminus V(G)$, then b is a sink by construction. Now, if (v_ℓ, v_k) is an arc, then it is added to D_k in the k^{th} iteration, where v_ℓ is a vertex in C_1, a clique in $\mathcal{G}_k$. Since v_k is the vertex of least index in C_1, $k < \ell$. □

Proposition 4.26. *The generalized elimination procedure for the phylogeny number produces an acyclic digraph D such that the phylogeny graph $P(D)$ has an induced subgraph isomorphic to G.*

Proof. Let G be a graph, $\mathcal{O} = v_1, v_2, \ldots, v_n$ an ordering of the vertices of G, and $\mathcal{G}$ an edge clique covering of G. From the initialization, the vertices of G are a subset of the vertices of D. Let v_k and v_ℓ, $k < \ell$, be vertices of D that are also vertices of G. Suppose that v_k and v_ℓ are adjacent in G. Let i be the least index such that $\mathcal{G}_i$ contains a clique C that contains the edge $\{v_k, v_\ell\}$. Since $\mathcal{G}$ is an edge clique cover of G, i is well-defined. Now if $C = C_1 \in \mathcal{G}_i$, then both the arcs (v_k, v_i) and (v_ℓ, v_i) are added to D_i in the i^{th} iteration, or if $i = k$, only the arc (v_ℓ, v_k) is added. Thus, v_k and v_ℓ are adjacent in $P(D)$. Otherwise, the arcs (v_k, b_j) and (v_ℓ, b_j) are added to D_i for some b_j, and again v_k and v_ℓ are adjacent in $P(D)$.

Suppose that v_k and v_ℓ are adjacent in $P(D)$. If v_k and v_ℓ have an arc connecting them in D, then by Lemma 4.25, the arc is oriented towards v_k. Thus, in the k^{th} iteration, $v_\ell \in C_1$ for a clique $C_1 \in \mathcal{G}_k$. Since both v_k and v_ℓ are in C_1, v_k and v_ℓ must be adjacent in G. Now, if v_k and v_ℓ have incident arcs oriented towards a common vertex x, where $x \neq v_k, v_\ell$, then these arcs are added in some i^{th} iteration of the procedure. Then both v_k and v_ℓ are in the same clique $C_j \in \mathcal{G}_i$, and so must be adjacent in G. □

Note that Remark 4.4 still holds in the phylogeny number case. We also have the following lemma and proposition, whose proofs are similar to the proofs of Lemma 4.6 and Proposition 4.7.

Lemma 4.27. *Let $\mathcal{E}_i$ be the edge coverings generated by the elimination procedure for the phylogeny number for a graph G and a vertex ordering $\mathcal{O} = v_1, \ldots, v_n$. Then the set $\mathcal{E} = \bigcup_{i=1}^{n} \mathcal{E}_i$ is an edge clique covering of G.*

Proposition 4.28. *Let G be a graph and $\mathcal{O} = v_1, \ldots, v_n$ be an ordering of the vertices of G. Then the number of vertices added to the digraph produced by the elimination procedure for the phylogeny number is the number of vertices added to the digraph produced by the generalized elimination procedure for the phylogeny number if the edge clique covering $\mathcal{G}$ is chosen to be $\mathcal{E}$ as defined in Lemma 4.27.*

Definition 4.29. The *phylogeny elimination number* $e_p(G, \mathcal{O}, \mathcal{G})$ of a graph G, an ordering $\mathcal{O}$ of the vertices, and an edge clique covering $\mathcal{G}$ is the number of vertices added to D so that $P(D)$ has G as an induced subgraph. Here D is the digraph produced by the generalized elimination procedure for the phylogeny number with G, $\mathcal{O}$, and $\mathcal{G}$ as inputs.

Definition 4.30. Let G be a graph, $\mathcal{O} = v_1, v_2, \ldots, v_n$ be an ordering of the vertices of G, and $\mathcal{G} = \{C_1, C_2, \ldots, C_k\}$ be an edge clique covering of G. For each vertex v_i, let $\mathcal{G}_i$ be the subset of $\mathcal{G}$ where for each $C_j \in \mathcal{G}_i$, v_i is the vertex in C_j of least index. Define

$$f_G(\mathcal{G}, \mathcal{O}) = \sum_{i=1}^{n} \max\{|\mathcal{G}_i| - 1, 0\}.$$

Lemma 4.31. *Let G be a graph, $\mathcal{O} = v_1, v_2, \ldots, v_n$ be an ordering of the vertices of G, and $\mathcal{G} = \{C_1, C_2, \ldots, C_k\}$ be an edge clique covering of G. Then $e_p(G, \mathcal{O}, \mathcal{G}) = f_G(\mathcal{G}, \mathcal{O})$.*

Proof. Note that $\mathcal{G}_i$ is defined exactly the same in both the generalized elimination procedure and in Definition 4.30. Note that in the i^{th} iteration, if $\mathcal{G}_i$ is empty, no arcs or vertices are added to D_i. If $\mathcal{G}_i$ is not empty, then $|\mathcal{G}_i| - 1$ new vertices are added as sinks to D_i. Thus, in the i^{th} iteration, $\max\{|\mathcal{G}_i| - 1, 0\}$ vertices are added to D_i, and, summing over all iterations,

$$e_p(G, \mathcal{O}, \mathcal{G}) = |V(D) \setminus V(G)| = \sum_{i=1}^{n} \max\{|\mathcal{G}_i| - 1, 0\} = f_G(\mathcal{G}, \mathcal{O}).$$

□

Lemma 4.32. *For a graph G, the phylogeny number $p(G)$ equals $\min_{\mathcal{G}} \min_{\mathcal{O}} f_G(\mathcal{G}, \mathcal{O})$, where $\mathcal{G}$ ranges over all edge clique coverings of G, and $\mathcal{O}$ ranges over all orderings of the vertices of G.*

Proof. Let $\mathcal{G} = \{C_1, C_2, \ldots, C_k\}$ be an edge clique covering of G, and $\mathcal{O} = v_1, v_2, \ldots, v_n$ be an ordering of the vertices of G. By Lemmas 4.25 and 4.26, the generalized elimination procedure produces an acyclic digraph D such that $P(D)$ has an induced subgraph isomorphic to G. By Lemma 4.31, $e_p(G, \mathcal{O}, \mathcal{G}) = f_G(\mathcal{G}, \mathcal{O})$, and so $p(G) \leq \min_{\mathcal{G}} \min_{\mathcal{O}} f_G(\mathcal{G}, \mathcal{O})$.

Now let F be an acyclic digraph that attains the phylogeny number for G; that is, $P(F)$ has an induced copy of G and $|V(F) \setminus V(G)| = p(G)$. Let $\mathcal{O} = v_1, v_2, \ldots, v_n$ be an ordering of the vertices of G such that if (v_ℓ, v_k) is an arc in F, then $k < \ell$. We construct an edge clique covering $\mathcal{G}$ of G from F as follows: For a vertex $v_i \in V(G)$, $N_F^{\text{in}}[v_i]$ induces a clique in G, and for a vertex $b \in V(F) \setminus V(G)$, $N_F^{\text{in}}(b)$ induces a clique in G. Since these are the only two ways edges can be present in G, $\mathcal{G}$ is an edge clique cover of G. Now observe that the digraph D produced by the generalized elimination procedure with $\mathcal{G}$ and $\mathcal{O}$ has the same number of vertices as F. In fact, if $C_1 \in \mathcal{G}_i$ is chosen to be the clique induced by $N_F^{\text{in}}[v_i]$, then D is isomorphic to F.

Therefore, $p(G) = |V(F) \setminus V(G)| = |V(D) \setminus V(G)| = e_p(G, \mathcal{O}, \mathcal{G}) \geq \min_{\mathcal{G}} \min_{\mathcal{O}} f_G(\mathcal{G}, \mathcal{O})$, and so $p(G) = \min_{\mathcal{G}} \min_{\mathcal{O}} f_G(\mathcal{G}, \mathcal{O})$. □

With our formula for evaluating different edge clique covers in hand, we can again turn our attention to kite-free graphs. The following lemmas are analogous to Lemmas 4.15, 4.17, and 4.18, and their proofs are the same.

Lemma 4.33. *Let G be a kite-free graph and $\mathcal{O} = v_1, \ldots, v_n$ be an ordering of the vertices of G. In the elimination procedure for the phylogeny number, an edge $\{v_j, v_k\}$ with $v_j, v_k \in N_G(v_i)$ appears in some clique of $\mathcal{E}_\ell$, where $\ell \leq i$.*

Lemma 4.34. *Let G be a graph, $\mathcal{O} = v_1, \ldots, v_n$ be an ordering of the vertices of G, and $\mathcal{G}$ be an edge clique covering of G. Then the cliques of $\mathcal{G}_i$ must cover $\widehat{G}_i$.*

Lemma 4.35. *Let G be a kite-free graph and $\mathcal{O} = v_1, \ldots, v_n$ be an ordering of the vertices of G. Then the subgraphs G'_i of G generated by the elimination procedure for the phylogeny number are exactly $\widehat{G}_i$.*

Thus we have

Lemma 4.36. *Let G be a kite-free graph and $\mathcal{O} = v_1, \ldots, v_n$ be an ordering of the vertices of G. Then $f_G(\mathcal{E}, \mathcal{O}) = \min_{\mathcal{G}} f_G(\mathcal{G}, \mathcal{O})$, where $\mathcal{E}$ is any edge clique cover produced by the elimination procedure for the phylogeny number on G and $\mathcal{O}$, and where the right-hand side minimum is taken over all edge clique covers $\mathcal{G}$ of G.*

Proof. Let $\mathcal{G}$ be an edge clique cover of G such that $f_G(\mathcal{G}, \mathcal{O})$ is minimized. By Lemma 4.34, $\mathcal{G}_i$ must cover $\widehat{G}_i$ for all i. But by Lemma 4.35, $G'_i = \widehat{G}_i$. Since $\mathcal{E}_i$ is chosen to be a minimum size cover of G'_i, $|\mathcal{E}_i| \leq |\mathcal{G}_i|$. Thus, $\max\{|\mathcal{E}_i| - 1, 0\} \leq \max\{|\mathcal{G}_i| - 1, 0\}$, and summing over i gives $f_G(\mathcal{E}, \mathcal{O}) \leq f_G(\mathcal{G}, \mathcal{O})$. □

We now specialize the definition of the elimination number.

Definition 4.37. Given a graph G and an ordering $\mathcal{O}$, let $\mathcal{E} = \mathcal{E}(G, \mathcal{O}) = \{\mathcal{E}_1, \mathcal{E}_2, \ldots, \mathcal{E}_n\}$ be edge clique coverings obtained during the elimination procedure for the phylogeny number. Again, the notation is ambiguous since the way to choose the $\mathcal{E}_i$ is not completely specified in the procedure. The *phylogeny elimination number* $e_p(G)$ is the minimum of $e_p(G, \mathcal{O}, \mathcal{E})$ over all orders $\mathcal{O}$ and some $\mathcal{E}$ obtained when using $\mathcal{O}$. We will show that for certain classes of graphs, if $e_p(G)$ is this minimum and is attained for $\mathcal{O}$ and some $\mathcal{E}$, then it is attained for the same $\mathcal{O}$ and any $\mathcal{E}$ corresponding to $\mathcal{O}$. If this is the case, $e_p(G)$ is unambiguously defined.

As with $M(G)$, the determination of necessary and sufficient conditions for $e_p(G)$ to be unambiguously defined is an interesting open problem. Lemma 4.32 shows that there is always a "right" clique cover for each order such that the minimum over orders attains the phylogeny number $p(G)$. Analogous to Kim and Roberts' result, we show that for the kite-free graphs $e_p(G)$ is unambiguously defined and equals $p(G)$. We also show that the same graph L in Figure 4.2 has the property that for each order $\mathcal{O}$ there is a choice of clique cover $\mathcal{E}_i$ in the elimination procedure for the phylogeny number such that $M(G, \mathcal{O}, \mathcal{E}) > k(G)$. This demonstrates that $e_p(G)$ is not unambiguously defined and is not equal to $p(G)$ for all graphs.

Theorem 4.38. *For a kite-free graph G, the phylogeny elimination number $e_p(G)$ is unambiguously defined and equals the phylogeny number $p(G)$.*

Proof. By Lemma 4.32, $p(G) = \min_{\mathcal{G}} \min_{\mathcal{O}} f_G(\mathcal{G}, \mathcal{O})$, where $\mathcal{G}$ ranges over all edge clique coverings of G and $\mathcal{O}$ ranges over all orderings of the vertices of G. By Lemma 4.31, $e_p(G, \mathcal{O}, \mathcal{E}) = f_G(\mathcal{E}, \mathcal{O})$ for any $\mathcal{E}$ corresponding to $\mathcal{O}$. By Lemma 4.36, $f_G(\mathcal{E}, \mathcal{O}) = f_G(\mathcal{E}', \mathcal{O})$ for any $\mathcal{E}$ and $\mathcal{E}'$ produced by the elimination procedure for the phylogeny number and corresponding to $\mathcal{O}$. It follows that $e_p(G, \mathcal{O}, \mathcal{E})$ and therefore the phylogeny elimination number $e_p(G)$ is unambiguously defined. Moreover, by Lemma 4.36, $e_p(G) = \min_{\mathcal{O}} e_p(G, \mathcal{O}, \mathcal{E}) = \min_{\mathcal{O}} \min_{\mathcal{G}} f_G(\mathcal{G}, \mathcal{O})$, and therefore, $e_p(G) = p(G)$. □

Because of the similarities in the elimination procedures for the competition and phylogeny numbers, the same graph L in Figure 4.2 is also shows that the elimination procedure for the phylogeny number does not always attain $p(G)$. Both of the following propositions are proved in a fashion similar to Propositions 4.21 and 4.22 above.

Proposition 4.39. *For each ordering $\mathcal{O}$ of the vertices of the graph L in Figure 4.2, there is a choice of edge clique coverings $\mathcal{E}_i$ such that the number of added vertices by the elimination procedure for the phylogeny number is greater than* 1.

Proposition 4.40. *The phylogeny number of the graph L in Figure 4.2 is* 1.

4.6 Open Problems

Many questions still exist about the existence and efficacy of elimination procedures that calculate the competition number or the phylogeny number of a graph. Despite the existence of the graph L in Figure 4.2, the Kim-Roberts elimination procedure is still of interest, particularly in determining for which graphs the procedure calculates $k(G)$. For instance, is L the smallest graph where the elimination procedure fails, or is there a smaller example? Is there an example with only one kite? Kites without the xy edge do not always admit a choice in clique covers. Is the elimination procedure exact when there is no choice? A complete characterization of when the procedure is optimal and when it is not is still open.

Of course, all of the above questions also apply to the elimination procedure for the phylogeny number. The calculation of the competition number and the phylogeny number both are essentially problems about finding a minimum-size edge clique cover, where the "size" of the cover is computed in different ways. Because of the similarities in the problems, as well as the success of applying techniques to both problems, it seems that a reduction from one problem to another should be possible. This would eliminate the necessity of checking techniques on both problems. In fact, the phylogeny graph of an acyclic digraph D is just the competition graph of D with loops added to each vertex. However, loops are normally excluded from digraphs when considering competition graphs, and so this reduction is not very useful. No reduction is known using only acyclic digraphs. One complicating factor is that available prey are a global property in the competition number case, whereas in the phylogeny number case, the number of vertices needed to be added is strictly a local property.

The existence of the graph L where both the Kim-Roberts elimination procedure and the elimination procedure for the phylogeny number fail suggests a natural question: Can a different elimination procedure be created that succeeds for all graphs? To answer this question, a more strict definition of what constitutes an elimination procedure is needed. One reasonable condition might be to restrict what portion of the graph the procedure may consider when eliminating a vertex v. For instance, the procedure might only be able to

consider vertices that are a fixed distance from v. In such instances where an elimination procedure can only consider local information, it seems unlikely that the procedure will calculate $k(G)$ or $p(G)$ for all graphs, even with the power of taking a minimum over all vertex orders. One indication supporting this view would be if it can be shown that the Kim-Roberts elimination procedure needs to solve an NP-complete problem about cliques to guarantee producing the competition number, despite the extra power of the minimum over orders. It might be possible to prove the NP-completeness using the "widgetlike" construction of the graph L. Another reasonable condition is requiring that all computation for eliminating a vertex is done in polynomial time. It seems in this case that examining factorial number of different vertex orders should give sufficient power to exactly solve either problem. However, an explicit procedure that accomplishes this is still needed.

A more general study of elimination procedures might also give insight into what graph parameters could be effectively calculated using elimination properties. Other parameters related to clique coverings are natural candidates, but perhaps other parameters such as chromatic number could also be considered.

There are many important open questions in the theory of competition graphs that this chapter has not touched on. We mention two long-standing problems here. An *interval graph* is a graph whose vertices can be associated with intervals in the real line such that two vertices are adjacent if and only if the associated intervals intersect. While investigating food webs that occur in nature, Cohen [1] empirically observed that a large number of food webs are interval graphs. If this observation is true for all food webs, the ecological implication is that there is one linear parameter (such as pH or temperature) for each ecosystem that determines the competitive relationships between species. It is not true that all competition graphs are interval graphs. However, a large literature has developed attempting to explain why competition graphs of food webs that occur in nature are often interval graphs. Of particular interest is determining which structural properties of an acyclic digraph guarantee that its competition graph is an interval graph. See [10] for a survey of these results.

A second major open problem involves a conjecture of Opsut. Opsut showed in [7] that the competition number of all line graphs is at most 2. Based on this result, Opsut conjectured that if the closed neighborhood of every vertex in a graph G can be edge-covered by at most two cliques of G, then the competition number of G is at most 2. Note that line graphs satisfy this hypothesis. Substantial progress has been made by Wang in [16], [17], and [18] towards this conjecture, but it still remains unresolved.

References

[1] J. E. Cohen, Interval Graphs and Food Webs: A Finding and a Problem, RAND Corporation Document 17696-PR, Santa Monica, CA, 1968.

[2] M. C. Golumbic, A Note on Perfect Gaussian Elimination, *J. Math. Anal. Appl.*, 64 (1978), 455-457.

[3] S.-R. Kim, The Competition Number and its Variants, in: *Quo Vadis Graph Theory?*, J. Gimbel, J. W. Kennedy, and L. V. Quintas, eds., Annals of Discrete Mathematics, 55 (1993), 313-325.

[4] S.-R. Kim and F. S. Roberts, Competition Numbers of Graphs with a Small Number of Triangles, *Discrete Applied Math.*, 78 (1997), 153-162.

[5] S.-R. Kim and F. S. Roberts, The Elimination Procedure for the Competition Number, *Ars Combinatoria,* 50 (1998), 97-113.

[6] J. R. Lundgren, Food Webs, Competition Graphs, Competition-Common Enemy Graphs, and Niche Graphs, in F. S. Roberts (ed.), *Applications of Combinatorics and Graph Theory in the Biological and Social Sciences*, vol 17 of *IMA Volumes in Mathematics and its Applications*, Springer-Verlag, New York, 1989, 221-243.

[7] R. J. Opsut, On the Computation of the Competition Number of a Graph, *SIAM J. Alg. & Discr. Meth.*, 3 (1982), 420-428.

[8] S. Parter, The Use of Linear Graphs in Gauss Elimination, *SIAM Review*, 3 (1961), 119-130.

[9] A. Raychaudhuri and F. S. Roberts, Generalized Competition Graphs and their Applications, in P. Brucker and R. Pauly (eds.), *Methods of Operations Research*, Academic Press, New York and London, 1969, 295-311.

[10] F. S. Roberts, Competition Graphs and Phylogeny Graphs, in L. Lovasz (ed.), *Graph Theory and Combinatorial Biology, Bolyai Society Mathematical Studies,* 7 (1999), *J. Bolyai Mathematical Society,* Budapest, Hungary, 333-362.

[11] F. S. Roberts, Food Webs, Competition Graphs, and the Boxicity of Ecological Phase Space, in Y. Alavi and D. Lick (eds.), *Theory and Applications of Graphs*, Springer, New York, 1978, 477-490.

[12] F. S. Roberts, *Graph Theory and its Applications to Problems of Society*, CBMS-NSF Monograph No. 29, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, 1978.

[13] F. S. Roberts and L. Sheng, Phylogeny Numbers, *Discrete Applied Mathematics*, 87 (1998), 213-228.

[14] F. S. Roberts and L. Sheng, Phylogeny Numbers of Graphs with Two Triangles, *Discrete Applied Math.*, 103 (2000), 191-207.

[15] D. J. Rose, Triangulated Graphs and the Elimination Process, *J. Math. Anal. Appl.*, 32 (1970), 597-607.

[16] C. Wang, *Competition Graphs, Threshold Graphs and Threshold Boolean Functions*, Ph.D. thesis, RUTCOR-Rutgers Center for Operations Research, Rutgers University, New Brunswick, NJ, 1991.

[17] C. Wang, On Critical Graphs for Opsut's Conjecture, *Ars Combinatoria*, 34 (1992), 183-203.

[18] C. Wang, Competitive Inheritance and Limitedness of Graphs, *J. Graph Theory*, 19 (1995), 353-366.

Vita

Stephen G. Hartke

1991-95 Attended St. Xavier High School, Cincinnati, Ohio. Graduated as valedictorian.

1995-99 Attended the University of Dayton, Dayton, Ohio. Graduated *summa cum laude* with a B.S. in Mathematics and Computer Science and minors in Physics and English.

1999-2004 Attended Rutgers University, Piscataway, New Jersey, for graduate work in Mathematics.

1999 (with A. W. Higgins) Maximum Degree Growth of the Iterated Line Graph, *The Electronic Journal of Combinatorics,* 6 (1999), #R28.

2000 Binary De Bruijn Cycles under Different Equivalence Relations, *Discrete Mathematics,* 215 (2000), 93-102.

2000 Summer Mathematics Research Experiences, *Proceedings of the Conference on Summer Undergraduate Mathematics Research Programs,* J. A. Gallian, ed., American Mathematical Society (2000), 267-271.

2002 (with D. Bianco, O. M. Carducci, and A. Larimer) Stable Matchings in the Couples Problem, *Morehead Electronic Journal of Applicable Mathematics*, Issue 2, MATH-2001-06, January 16, 2002.

2003 (with A. W. Higgins) Minimum Degree Growth of the Iterated Line Graph, *Ars Combinatoria*, 69 (2003), 275-283.

2003 (with M. Develin and D. P. Moulton) A General Notion of Visibility Graphs, *Discrete and Computational Geometry*, 29 (2003), no. 4, 511-524.

2003 The Elimination Procedure for the Phylogeny Number, *Ars Combinatoria*, accepted for publication.

2003 (with W. S. Diestelkamp and R. H. Kenney) On the Degree of Local Permutation Polynomials, *Journal of Combinatorial Mathematics and Combinatorial Computation*, accepted for publication.

2003 The Elimination Procedure for the Competition Number is Not Optimal, DIMACS Technical Report 2003-42.