Optimization Policies for Polya Contagion Networks

by

Greg Harrington

A thesis submitted to the
Department of Mathematics and Statistics
in conformity with the requirements for
the degree of Master of Applied Science

Queen's University
Kingston, Ontario, Canada
June 2020

Copyright © Greg Harrington, 2020

Abstract

This thesis investigates optimization policies for resource distribution in network epidemics, using a model that derives from the classical Polya process. The basic mechanics of this model, called the network Polya process, are based on a modified urn sampling scheme that accounts for both temporal and spatial contagion between neighbouring nodes in a network. We present various infection metrics and use them to develop two optimization problems, one which takes place upon initialization and one which occurs continually as the network Polya process develops. We frame these problems as both one-sided and two-sided resource allocation problems with fixed budgets, and analyze a suite of potential policies. Due to the complexity of these problems, we introduce effective proxy measures for the average infection rate in each case. We also prove that the two-sided infection-curing game on the so-called expected network exposure admits a Nash equilibrium. In both the curing and initialization scenarios we introduce heuristic policies that primarily function on the basis of limiting the number of targeted nodes within a particular network setup. Simulations are run for large-scale networks to compare performance of our heuristics to provably convergent gradient descent algorithms run on the simplified proxy measures.

Acknowledgments

I would like to thank my supervisors Prof. Fady Alajaji and Prof. Bahman Gharesifard for their undying patience and support, and for providing for me this opportunity.

A sincere thanks as well to Prof. Tamas Linder, for reviewing this thesis and providing a much simpler proof for Lemma 3.1.5.

I would also like to extend my appreciation to both of my parents, whom I've never properly thanked for all that they've done for me.

Finally, I would like to thank my friends, both real and imaginary. I'm sure you're out there somewhere.

Contents

List of Figures

Chapter 1

Introduction

The study of epidemics on networks is an active research topic, e.g., see [10, 23, 26] and the references therein. Real-life examples include the propagation of burst errors in a wireless communication channel [2], of a biological disease through a population [21], of malware in computer or smartphone systems [13], and the dissemination of rumors [31] and competing opinions [1] in social networks.

In this thesis we examine one- and two-sided problems for the network Polya contagion process as introduced in [17, 18]. The setup for this model is an adaptation of the classical Polya contagion process [11, 29, 28] to allow for spatial interactions between neighbouring nodes. This model is similar to the well-known class of Susceptible-Infected-Susceptible (SIS) epidemics models [10]. Indeed in [18], it was empirically shown that the network Polya contagion model can mimic the behaviour of the SIS process. SIS models have been studied in a game-theoretic context in [20, 27] and references therein. We herein establish similar problems for the network Polya contagion model.

We begin with an overview of the classical Polya process, which is a temporal contagion process that evolves through a method of sampling from an urn containing

a finite number of red and black balls, representing "infected" and "healthy" states, respectively. The classical Polya model has been used in applications such as consensus dynamics [12] and generalized for various other purposes [8, 9]. The network Polya process modifies this sampling scheme by introducing "super urns", which are allocated for each network node and contain red and black balls from both a node's individual urn as well as those of neighbouring nodes. This modification introduces an element of spatial contagion not present in the classical model. We overview various metrics used to measure infection in the network Polya process; we then highlight two network models designed to simplify this new modified contagion porcess.

Working off of this basis, we develop two separate resource allocation problems. The first problem considers a one-time application of a control policy upon initialization of a Polya urn network, while the second concerns the continual distribution of resources as the Polya network process develops. Both of these problems are written in the context of trying to manipulate the "average network-wide infection rate". Chapter 3 concerns the setup and analysis of the first such problem, while Chapter 4 concerns the second.

The initialization problem works on the idea of one party controlling the distribution of resources within a network in an effort to minimize the average network-wide infection rate. Resources in this case concern the distribution of red and black balls within a network, prior to any draws taking place. We present multiple variations of this setup, but primarily focus on the one-sided finite horizon case, wherein a player controls the allocation of black balls within the network according to a fixed budget, with the goal of minimizing the average infection rate at some fixed point in the future.

We provide a series of results dedicated to showing that optimal policies for this problem will satisfy two conditions: nested nodes will receive no resources, and symmetric red ball initializations will yield symmetric black ball initializations (the specifics of what is meant by "symmetry" in this case is further detailed using fundamental ideas from graph theory). We provide empirical evidence to support these claims in cases where providing analytical guarantees is complicated. Since the metric used to measure the average infection rate becomes increasingly complex with time, we also introduce a simple one-step proxy measure to simplify this problem, and we prove that under certain conditions this proxy measure admits an optimal policy. An algorithm to obtain such an optimal policy using gradient descent is given and used as a base with which we compare our own heuristic policies. We also provide a brief analysis of this problem under the limitation of only being able to target a single network node. This consideration lends a hand in the development of our heuristics. We detail three varieties of heuristic policies, each of which works on the basis of limiting our set of viable target nodes. The first set of policies target the set of "inner nodes", the second works through a layered application of this inner node targeting technique, and the last works by iteratively targeting the most central nodes until full network coverage (either direct or indirect) is obtained. We provide two different algorithms for determining these target sets, and provide variations based on previously tested heuristics (see [19]).

For the second control problem, which we refer to as the "infection-curing problem" or the "Delta-curing problem", we first define a two-player game on the average infection rate for a given network. This setup is an extension of the contagion curing problem that was studied extensively in [19]. In order to simplify the problem of

finding optimal control policies, we consider a game on the proxy measure of the expected network exposure. We establish that the expected network exposure is convex as a function of the curing parameters and concave as a function of the infection parameters. We prove that under budget constraints, there exists a deterministic Nash equilibrium (see [24, 25]) for the game on the expected network exposure, which can be determined using gradient descent algorithms. Finally, we establish a set of heuristic policies for the Delta-curing problem that work using the same methods as in the initialization case.

In Chapter 5, we provide simulations to demonstrate the performance of the proposed heuristics for the initialization and curing problems on both sparse and dense large-scale networks. We start by illustrating the differences between the target sets generated using our previously established algorithms. We then run trials that empirically measure the average network-wide infection rate for the various heuristic policies. These policies are compared to optimal policies for the one-step proxy measures. The results of these simulations, in both the initialization and curing cases, are then summarized.

Chapter 2

Polya Network Contagion Process

2.1 Preliminaries

Throughout this thesis a number of results are derived using standard probability concepts, which can be found in various texts including [4, 15]. Herein, let $(\Omega, \mathcal{F}, P)$ be a probability space. Consider the stochastic process $\{Z_n\}_{n=1}^{\infty}$, where $Z_n = (Z_{1,n}, \ldots, Z_{N,n})$ is a random vector on Ω. Here we tend to think of the indices of this process as time indices, while the vector indices can be thought of as spatial indices (in this work, they are used to distinguish nodes within a network). We write the sequence of variables $(Z_{i,1}, \ldots, Z_{i,n})$ as Z_i^n, and more generally, we use the notation $Z_{i,s}^t = (Z_{i,s}, Z_{i,s+1}, \ldots, Z_{i,t})$, where $1 \leq s < t \leq n$.

Definition 2.1.1. *A sequence of σ-algebras $\{\mathcal{F}_n\}_{n=1}^{\infty}$ is a filtration on a process $\{Z_n\}_{n=1}^{\infty}$ if*

- *$\mathcal{F}_t \subset \mathcal{F}_n \subset \mathcal{F}$ for all $t \leq n$*
- *Z_n is measurable with respect to $\mathcal{F}_n$ for all n*

In general, we will consider the natural filtration $\{\mathcal{F}_n^Z\}_{n=1}^{\infty}$, which is essentially

the smallest filtration on the process $\{Z_n\}_{n=1}^{\infty}$; i.e., $\{\mathcal{F}_n^Z\}_{n=1}^{\infty} \subseteq \{\mathcal{F}_n\}_{n=1}^{\infty}$ for any other filtration $\{\mathcal{F}_n\}_{n=1}^{\infty}$ on $\{Z_n\}_{n=1}^{\infty}$. Alternatively, the natural filtration can be thought of as the sequence of σ-algebras generated by the stochastic process, wherein at each time step, the σ-algebra associated with that time step has all information about the stochastic process up to that point in time, but no additional information.

Throughout this work, we refer to Barabasi-Albert networks (graphs) [3], which are scale-free models generated through a method of preferential attachment. The resulting structure of these models have properties resembling real-world social networks, which make them a useful tool for analyzing the proposed control policies presented in this work.

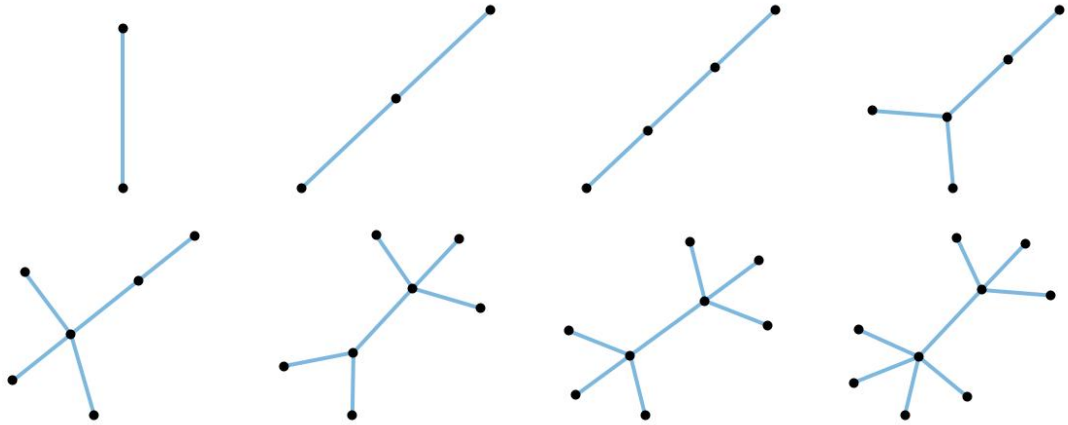


Figure 2.1: Sample demonstration of Barabasi-Albert network generation. This example begins with $m_0 = 2$ connected nodes, and then each added node is assigned $m = 1$ edge using a method of preferential attachment.

In order to generate a Barabasi-Albert network, we begin with a complete network of size m_0. Then, nodes are added iteratively until we arrive at the desired network size, with each added node being assigned a fixed number of edges $m < m_0$. These new edges will form randomly between the added node and the existing nodes in

the network, with a preference for existing nodes that have a greater number of connections. For a network with k nodes, the probability that the new node forms an edge with a given node i is given by by the formula $\frac{deg(i)}{\sum_{j=1}^{k} deg(j)}$, where $deg(i)$ is the degree, i.e., the number of edges, of node i. A demonstration of how this algorithm functions can be seen in Figure 2.1.

2.2 Classical Polya Process

We recall the classical Polya contagion process [11, 29, 28]. An urn initially contains $R \in \mathbb{Z}_{>0}$ red balls and $B \in \mathbb{Z}_{>0}$ black balls. We use $T = R + B$ to denote the total number of balls in the urn. At each time, n, a ball is drawn from the urn and returned with $\Delta > 0$ balls of the same colour. The random variable Z_n is used to indicate the colour of the ball on the nth draw, where

$$Z_n = \begin{cases} 1 & \text{if the } n\text{th draw is red} \\ 0 & \text{if the } n\text{th draw is black.} \end{cases}$$

We can then use U_n to denote the proportion of red balls in the urn after the nth draw, where

$$U_n := \frac{R + \Delta \sum_{t=1}^{n} Z_t}{T + n\Delta} = \frac{\rho_c + \delta_c \sum_{t=1}^{n} Z_t}{1 + n\delta_c}.$$

Here $\rho_c = \frac{R}{T}$ is the initial proportion of red balls in the urn and $\delta_c = \frac{\Delta}{T}$ is a correlation parameter. The conditional probability of drawing a red ball at time n , given the past history of draws $Z^{n-1} := (Z_1, \ldots, Z_{n-1})$, is given by

$$P(Z_n = 1 \mid Z^{n-1}) = \frac{R + \Delta \sum_{t=1}^{n-1} Z_t}{T + (n-1)\Delta} = U_{n-1}. \tag{2.1}$$

Additionally, we note that both U_n and the process sample average $\frac{1}{n} \sum_{t=1}^{n} Z_t$ converge almost surely (a.s.) to a random variable governed by a Beta distribution with

parameters $\frac{\rho_c}{\delta_c}$ and $\frac{1-\rho_c}{\delta_c}$; i.e., there exists $X \sim \text{Beta}(\frac{\rho_c}{\delta_c}, \frac{1-\rho_c}{\delta_c})$ on $(\Omega, \mathcal{F}, P)$, such that

$$U_n \xrightarrow{a.s.} X \qquad \text{and} \qquad \frac{1}{n}\sum_{i=1}^{n} Z_i \xrightarrow{a.s.} X \qquad \text{as } n \to \infty.$$

The classical Polya process $\{Z_n\}_{n=1}^{\infty}$ is fully described by the parameters δ_c and ρ_c, and is denoted by Polya(ρ_c,δ_c). This process is exchangeable (thus stationary) and non-ergodic, and has a 1-dimensional distribution given by $P(Z_n = a) = (\rho_c)^a(1-\rho_c)^{1-a}$, $a \in \{0, 1\}$. Variations on this model have been explored in a number of other works. We focus exclusively on an adaptation of this model first introduced in [17] and further explored in [18],[19], designed as a network process that introduces spatial contagion to the classical temporal contagion process presented above.

2.3 Network Polya Process

We now review the network Polya contagion process introduced in [17, 18]. For the most part, we try herein to maintain the conventions as defined in those works. Let $\mathcal{G} = (V, \mathcal{E})$ be a graph, where $V = \{1, \ldots, N\}$ is the set of $N \in \mathbb{Z}_{>0}$ nodes and $\mathcal{E} \subset V \times V$ is the set of edges. Throughout, $\mathcal{G}$ is taken to be undirected (i.e., $(i, j) \in \mathcal{E}$ if and only if $(j, i) \in \mathcal{E}$ for all $i, j \in V$), and is assumed to be connected (i.e., there exists a series of edges connecting any two nodes in $\mathcal{G}$). The set of neighbours to node i is given by $\mathcal{N}_i = \{v \in V : (i, v) \in \mathcal{E}\}$. We then define $\mathcal{N}_i' = \{i\} \cup \mathcal{N}_i$. Each node $i \in V$ is assigned an urn with $R_i \in \mathbb{Z}_{>0}$ red balls and $B_i \in \mathbb{Z}_{>0}$ black balls, with the total number of balls at node i denoted by $T_i = R_i + B_i$. We denote the black and red ball initializations by the vectors $\mathbf{B} := (B_1, \ldots, B_N)$ and $\mathbf{R} := (R_1, \ldots, R_N)$, respectively. We will refer to $\mathbf{B}$ as the curing initialization and $\mathbf{R}$ as the infection initialization.

We next introduce the concept of a super urn for a given node i, which contains

the red and black balls at node i as well as the red and black balls of each of its neighbouring nodes. This concept is illustrated in Figure 2.2. For each node $i \in V$, we use $\bar{R}_i = \sum_{j \in \mathcal{N}_i'} R_j$ and $\bar{B}_i = \sum_{j \in \mathcal{N}_i'} B_j$ to denote the number of red and black balls respectively in node i's super urn. The total number of balls in the ith super urn is given by $\bar{T}_i = \bar{R}_i + \bar{B}_i$. Similar to the classical Polya process, a draw is conducted for each node at each time, and then a number of balls of the same colour is added to that node's individual urn. In this case, however, the draw is conducted on the super urn of each node. As well, we may allow for the number of added balls to vary based on which node the draw was for, the colour of the draw, and the time at which the draw occurred. Thus, for node i at time n, if a red ball is drawn we add $\Delta_{r,i}(n)$ red balls to node i's individual urn and if a black ball is drawn we add $\Delta_{b,i}(n)$ black balls to node i's individual urn. We use an indicator, $Z_{i,n}$ to denote the colour of the nth draw for node i, where

$$Z_{i,n} = \begin{cases} 1 & \text{if the } n\text{th draw for node } i \text{ is red} \\ 0 & \text{if the } n\text{th draw for node } i \text{ is black.} \end{cases}$$

We refer to $\{\Delta_{b,i}(n)\}_{n=1}^{\infty}$ as the curing parameters and $\{\Delta_{r,i}(n)\}_{n=1}^{\infty}$ as the infection parameters for node i. From an urn-sampling point of view these Δ's are nonnegative integers; however, we allow them to be nonnegative real-valued numbers for mathematical convenience (e.g., when determining optimal control policies). We will likewise do the same for the initialization parameters $\mathbf{B}$ and $\mathbf{R}$.

Ultimately, the collection of draw values at each time step form our network Polya contagion process, represented as $\{Z_n\}_{n=1}^{\infty}$, where Z_n represents the vector $(Z_{1,n}, \ldots, Z_{N,n})$. If desired, we can also separate out the individual draw process $\{Z_{i,n}\}_{n=1}^{\infty}$ for a given node i. We are interested in analyzing how the network process

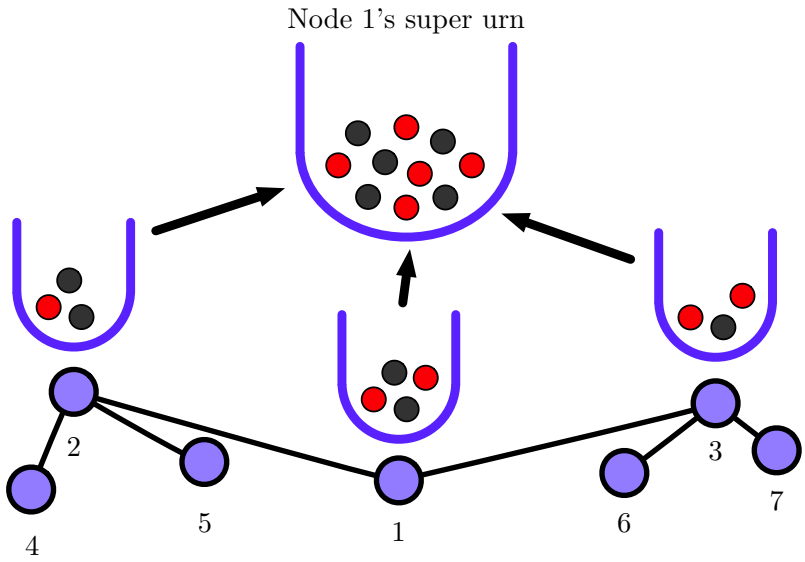


Figure 2.2: Illustration of a super urn in a network [16].

evolves in time, and implementing control policies on the infection and curing parameters where possible. We first introduce some metrics necessary for this analysis.

Similar to the classical Polya process, we use $U_{i,n}$ to denote the proportion of red balls in node i's urn after the nth draw. Letting

$$X_{i,n} = T_i + \sum_{t=1}^{n} \Delta_{r,i}(t) Z_{i,t} + \Delta_{b,i}(t)(1 - Z_{i,t}) \tag{2.2}$$

represent the total number of balls in node i's urn after the nth draw, we can write

$$U_{i,n} = \frac{R_i + \sum_{t=1}^{n} \Delta_{r,i}(t) Z_{i,t}}{X_{i,n}}. \tag{2.3}$$

Here we note that $U_{i,0} = \frac{R_i}{T_i}$ represents the initial (at time $n = 0$) proportion of red balls in node i's individual urn. Though for the classical Polya process the individual urn proportion is commonly used in the analysis, in this case, the information we gain through this value is limited by the fact that our draws are performed on the super urns for each node, rather than the urns for the individual nodes themselves. We therefore also need a metric to measure the proportion of red balls in the super

urn of node i at time n, which we denote by $S_{i,n}$. This proportion is given by the equation

$$S_{i,n} = \frac{\bar{R}_i + \sum_{j\in\mathcal{N}_i'}\sum_{t=1}^{n}\Delta_{r,j}(t)Z_{j,t}}{\sum_{j\in\mathcal{N}_i'}X_{j,n}} = \frac{\sum_{j\in\mathcal{N}_i'}U_{j,n}X_{j,n}}{\sum_{j\in\mathcal{N}_i'}X_{j,n}}. \tag{2.4}$$

Like for the individual node proportions, at time $n = 0$ we use the convention $S_{i,0} = \frac{\bar{R}_i}{\bar{T}_i}$. This is simply representative of the proportion of red balls in node i's super urn when the process is initialized. From these two sets of values we define

$$\tilde{U}_n = \frac{1}{N}\sum_{i=1}^{N}U_{i,n},$$

which we refer to as the *network susceptibility*, and

$$\tilde{S}_n = \frac{1}{N}\sum_{i=1}^{N}S_{i,n},$$

which we refer to as the *network exposure*. Both the network susceptibility and network exposure are functions of the underlying network contagion process $\{Z_n\}_{n=1}^{\infty}$, as are the respective measures for each node. In general, however, we will omit these arguments when making reference to a particular function, unless otherwise noted. For example, for a given node i, we will write $S_{i,n}$ rather than $S_{i,n}(\{Z_j^n\}_{j=1}^N)$, and likewise for $U_{i,n}$. This is done for the sake of readability and notational ease.

Next we calculate infection probabilities within the network; i.e., the likelihood of drawing red balls. To start, the conditional probability of drawing a red ball for node i at time n , given the past history of draws for all the nodes

$$\{Z_j^{n-1}\}_{j=1}^N := \{(Z_{1,1},\ldots,Z_{1,n-1}),\ldots(Z_{N,1},\ldots,Z_{N,n-1})\},$$

is given by

$$P\left(Z_{i,n}=1|\{Z_j^{n-1}\}_{j=1}^N\right) = \frac{\bar{R}_i + \sum_{j\in\mathcal{N}_i'}\sum_{t=1}^{n-1}\Delta_{r,j}(t)Z_{j,t}}{\sum_{j\in\mathcal{N}_i'}X_{j,n-1}} = S_{i,n-1}. \tag{2.5}$$

This is analogous to (2.1) of the classical Polya process, except that for the network

contagion process this conditional probability is represented by the super urn proportion rather than the individual urn proportion. We also need an unconditioned version of this equation. The n-fold joint probability of the network $\mathcal{G}$, which is a function defined for each specific draw history $a_i^n \in \{0,1\}^n$, $i \in V$, can be written as

$$\begin{aligned} P_{\mathcal{G}}^{(n)}(a_1^n, \ldots, a_N^n) &:= P(\{Z_i^n = a_i^n\}_{i=1}^N) \\ &= \prod_{t=1}^{n} P(\{Z_{i,t} = a_{i,t}\}_{i=1}^N | \{Z_i^{t-1} = a_i^{t-1}\}_{i=1}^N) \\ &= \prod_{t=1}^{n} \prod_{i=1}^{N} (S_{i,t-1})^{a_{i,t}} (1 - S_{i,t-1})^{1-a_{i,t}}. \end{aligned} \tag{2.6}$$

It is important to note here that the draw value of different nodes at a given time t are independent when conditioned on the history of the process up to that point, thus allowing us to write $P_{\mathcal{G}}^{(n)}$ in the form above. Finally, we introduce the metric

$$\tilde{I}_n := \frac{1}{N} \sum_{i=1}^{N} P(Z_{i,n} = 1), \tag{2.7}$$

which we call the *average infection rate at time n*. Since $P(Z_{i,n} = 1)$ represents the marginal probability that a node i is infected at time n, the average infection rate gives us the network-wide average of seeing infection within a node. Thus, this value represents the marginal probability that a randomly selected node within the network is infected at time n. In most cases where we look to study the asymptotic behaviour of the network contagion process, we do so in regard to this metric.

We can further break down Equation (2.7) by noting that

$$\begin{aligned} P(Z_{i,n} = 1) &= \sum_{\{a_j^{n-1}\}_{j=1}^N} P(Z_{i,n} = 1 | \{Z_j^{n-1} = a_j^{n-1}\}_{j=1}^N) P(\{Z_j^{n-1} = a_j^{n-1}\}_{j=1}^N) \\ &= \sum_{\{a_j^{n-1}\}_{j=1}^N} S_{i,n-1}(\{a_j^{n-1}\}_{j=1}^N) P_{\mathcal{G}}^{(n-1)}(\{a_j^{n-1}\}_{j=1}^N), \end{aligned} \tag{2.8}$$

where the last summations are over all possible draw histories $a_j^{n-1} \in \{0,1\}^{n-1}$, $j \in V$.

Each of these measures $\tilde{U}_n$, $\tilde{S}_n$, and $\tilde{I}_n$ are closely related to one another and can be useful in various situations. However, all of these values are network-specific as in general all three depend in some way on the underlying network topology and the initial ball distributions. The quantity $\tilde{I}_n$ is particularly difficult to analyze, and so more often than not, we resort to using alternative measures that allow us to simplify our analysis or gain insight into how $\tilde{I}_n$ behaves.

Looking at Equations (2.3) and (2.4) (and just from general intuition) it is clear that an increase in the node-specific proportion of red balls increases the super urn proportion of red balls in all neighbouring nodes. Likewise, if we consider how Equation (2.5) relates to Equation (2.8) it would make sense that an increase in the super urn proportion would also lead to an increase in the marginal network-wide infection rate; however, we will go into this further in the following chapters. For now, we have that

$$\uparrow U_{i,n} \implies \uparrow S_{j,n} \text{ for all } j \in \mathcal{N}_i',$$

and since $\tilde{I}_n$ is essentially the marginal version of the conditional $\tilde{S}_n$, we often start our analysis by using $\tilde{S}_n$ in place of $\tilde{I}_n$, without losing significant information.

2.3.1 Network Models

We next establish some network models that allow us to better understand how the network contagion process evolves. These models are described in detail in [16], but for our purposes, we provide just a general overview.

We have two models which we can use to approximate the behaviour of the Polya network contagion process, both of which are based on the classical Polya process. They are described as follows:

Model I. (Large-Network Analytical Model): *For a given node i, we approximate the evolution of that node's individual contagion process $\{Z_{i,n}\}_{n=1}^{\infty}$ using a classical Polya process, Polya$(\rho_{c,i}, \delta_{c,i})$ with parameters*

$$\rho_{c,i} = \frac{\bar{R}_i}{\bar{T}_i} \qquad \textit{and} \qquad \delta_{c,i} = \frac{\Delta}{\bar{T}_i + (N-1)\Delta}.$$

Model II. (Small-Network Analytical Model): *For a given node i, we approximate the evolution of that node's individual contagion process $\{Z_{i,n}\}_{n=1}^{\infty}$ using a classical Polya process, Polya$(\rho_{c,i}, \delta_{c,i})$ with parameters*

$$\rho_{c,i} = \frac{\bar{R}_i}{\bar{T}_i} \qquad \textit{and} \qquad \delta_{c,i} = \frac{\Delta}{\bar{T}_i N + (N-1)\Delta}.$$

There are a couple of stipulations that come with both of these models. First, it is important to note that as indicated by their descriptions, these models assume a fixed value Δ number of balls of a single colour are added to each node after each draw, regardless of any other factors that might normally be considered. Secondly, as is further detailed in [16], these models are only valid approximations under certain conditions. Model I provides a good approximation for the limiting behaviour of the network process for larger networks, whereas Model II is more appropriate for smaller networks. Though rather simplified versions of a complex process, both may be useful under the right circumstances.

Chapter 3

Initialization Problems

When we consider the problem of limiting the spread of infection for the Polya urn network contagion model, there are different ways to implement a curing policy. One method is to control the allocation of our curing parameters $\{\Delta_{b,i}(n)\}_{i=1}^{N}$ at each time n, which we explore in Chapter 4. We can, alternatively, control how the process is initialized. This is where we currently draw our focus for discussion.

3.1 Black Ball Allocation Problem

The initialization parameters $\mathbf{B} \in \mathbb{R}_{\geq 0}^{N}$ and $\mathbf{R} \in \mathbb{R}_{\geq 0}^{N}$ can be tailored to alter the evolution of the Polya contagion process for a given network. The goal is to determine how to allocate such resources so as to achieve a more desirable result, and determine whether or not a theoretically optimal initial allocation policy exists. Such optimizations may be performed for either or both sets of the initialization parameters subject to an initialization budget $\mathcal{B} \in \mathbb{R}_{\geq 0}$. We start with a formal setup for the one-sided initialization problem, wherein we aim to minimize the average infection for a given time n over the black ball initialization parameters $\mathbf{B}$ subject to a budget $\mathcal{B}_b$. Throughout this section we assume the initial distribution of red balls $\mathbf{R}$ is fixed.

Problem 3.1.1. (One-Sided Finite Horizon Constrained Budget Initialization): *For a fixed time n, minimize the average infection rate $\tilde{I}_n$ subject to a budget $\mathcal{B}_b$ on the curing initialization $\mathbf{B} = (B_1, \ldots, B_N)$, i.e., find:*

$$\min_{\{B_i\}_{i=1}^N \in \mathbb{R}^N_{\geq 0} : \sum_{i=1}^N B_i \leq \mathcal{B}_b} \tilde{I}_n.$$

We could also establish a formal setup for the two-sided initialization problem, wherein we are again minimizing the average infection rate for a given n, but in this case we do so over both sets of initialization parameters $\mathbf{B}$ and $\mathbf{R}$. In this case, however, it is important to note that while $\mathbf{B}$ is subject to a maximum budget of $\mathcal{B}_b$, we must implement a minimum budget $\mathcal{B}_r$ for $\mathbf{R}$, otherwise, the problem becomes trivial. We can elect to choose whether or not these budgets are equal.

Problem 3.1.2. (Two-Sided Finite Horizon Constrained Budget Initialization): *For a fixed time n, minimize the average infection rate $\tilde{I}_n$ subject to a maximum budget $\mathcal{B}_b$ on the curing initialization $\mathbf{B} = (B_1, \ldots, B_N)$ and a minimum budget $\mathcal{B}_r$ on the infection initialization $\mathbf{R} = (R_1, \ldots, R_N)$, i.e., find:*

$$\min_{\{B_i\}_{i=1}^N \times \{R_i\}_{i=1}^N \in \mathbb{R}^{2N}_{\geq 0} : \sum_{i=1}^N B_i \leq \mathcal{B}_b, \sum_{i=1}^N R_i \geq \mathcal{B}_r} \tilde{I}_n.$$

Both of these problems can also be extended to consider the inifinite horizon case, but for the purposes of our work, we will limit ourselves to the finite horizon. We can also note that while both problems may be interesting to analyze, from a practical perspective Problem 3.1.2 does not make much sense, as generally an interested party would not have control over the allocation of both sets of initialization parameters. For that reason, we limit our focus to Problem 3.1.1. It is important to note that the solution for this problem in general will change depending on the choice of n, so it is not possible to find a closed form solution that works for all n. We can however,

find some simplifications. We can also note that due to the inherent symmetry in the network model, that an analogous set of problems can be established wherein we look to maximize the average infection rate. As such, for many of the results in this chapter, we will make reference to solutions for the infection-maximization cases without providing explicit proofs. We start by considering Problem 3.1.1 for $n = 1$. We have

$$\begin{aligned}\tilde{I}_1 &= \frac{1}{N}\sum_{i=1}^{N} P(Z_{i,1} = 1) \\ &= \frac{1}{N}\sum_{i=1}^{N} \frac{\bar{R}_i}{\bar{R}_i + \bar{B}_i}. \end{aligned} \tag{3.1}$$

We note that as a function of the initialization parameters, $\tilde{I}_1$ is well-defined over the space $\{(\mathbf{B}, \mathbf{R}) \in \mathbb{R}^{2N}_{\geq 0} | \bar{B}_i + \bar{R}_i \neq 0, \forall i \in V\}$. We can additionally note that over this space, $\tilde{I}_1$ is convex as a function of $\mathbf{B}$ and concave as a function of $\mathbf{R}$ (the proof of this is straightforward). We use the form of (3.1) to provide some simplifications regarding solutions to Problem 3.1.1 for $n = 1$. We will then try to extend these results to a general time n.

3.1.1 Outer Node Allocation

Lemma 3.1.3. *Consider a general network $\mathcal{G} = (V, \mathcal{E})$ equipped with the Polya network contagion model. If for any nodes $i, j \in V$ we have $\mathcal{N}_i' \subset \mathcal{N}_j'$, then it can be assumed that $B_i = 0$ when minimizing $\tilde{I}_1$ over the initial distribution of black balls. Likewise, we can assume $R_i = 0$ when maximizing $\tilde{I}_1$ over the initial distribution of red balls.*

Proof. Consider a given initial distribution of black balls $\mathbf{B} = (B_1, \ldots, B_N)$, and red

balls $\mathbf{R} = (R_1, \ldots, R_N)$. As written in (3.1), we have

$$\tilde{I}_1 = \frac{1}{N} \sum_{k=1}^{N} \frac{\bar{R}_k}{\bar{R}_k + \bar{B}_k}.$$

Suppose, without loss of generality that $\mathcal{N}_1' \subset \mathcal{N}_2'$, and let $\mathbf{B}^* = (0, B_1 + B_2, B_3, \ldots, B_N)$ be an alternative curing initialization. For a given node $k \in V$ we have 3 cases to consider:

1. $k \in \mathcal{N}_1' \implies k \in \mathcal{N}_2'$
2. $k \notin \mathcal{N}_1'$ and $k \in \mathcal{N}_2'$
3. $k \notin \mathcal{N}_2' \implies k \notin \mathcal{N}_1'$

For cases 1 and 3, we can note that $\sum_{l \in \mathcal{N}_k'} B_l = \sum_{l \in \mathcal{N}_k'} B_l^*$. For case 2, we have $\sum_{l \in \mathcal{N}_k'} B_l \leq \sum_{l \in \mathcal{N}_k'} B_l^*$ with equality iff $B_1 = 0$. Denoting $\bar{B}_k^* = \sum_{l \in \mathcal{N}_k'} B_l^*$ we can see that

$$\frac{\bar{R}_k}{\bar{R}_k + \bar{B}_k} \geq \frac{\bar{R}_k}{\bar{R}_k + \bar{B}_k^*}, \forall k \in V.$$

Therefore, when we consider $\tilde{I}_1$ as a function of the initial distribution of black balls we get that $\tilde{I}_1(\mathbf{B}^*) \leq \tilde{I}_1(\mathbf{B})$. The proof for the maximization case follows in a similar manner. □

Essentially, we are stating that for the $n = 1$ case of the one-sided initialization problem (Problem 3.1.1), nodes with a nested neighbourhood can be ignored. We refer to such nodes as "outer nodes" due to their topological location. In order to extend this idea to a general n, we provide some preliminary results.

Lemma 3.1.4. *Consider two sequences of draw values $\{a_j^n\}_{j=1}^N$ and $\{b_j^n\}_{j=1}^N$, where both sequences are equal, except for some $(k, s) \in V \times \{1, \ldots, n\}$ where $a_{k,s} = 1$ and*

$b_{k,s} = 0$. *Then* $P(Z_{i,n+1} = 1|\{Z_j^n = a_j^n\}_{j=1}^N) \geq P(Z_{i,n+1} = 1|\{Z_j^n = b_j^n\}_{j=1}^N)$, *for all* $i \in V$.

Proof. Assume that $k \in \mathcal{N}_i'$ (note that if this is not the case, the two values we wish to compare are equal). Let

$$y = \sum_{j \in \mathcal{N}_i'} \sum_{t=1}^{n} a_{j,t} \Delta_{r,j}(t)$$

$$x = \sum_{j \in \mathcal{N}_i'} \sum_{t=1}^{n} (1 - a_{j,t}) \Delta_{b,j}(t).$$

Then letting $y^* = y - \Delta_{r,k}(s)$, and $x^* = x + \Delta_{b,k}(s)$, we obtain

$$\begin{aligned} P(Z_{i,n+1} = 1|\{Z_j^n = a_j^n\}_{j=1}^N) &= \frac{\bar{R}_i + y}{\bar{R}_i + \bar{B}_i + y + x} \\ &\geq \frac{\bar{R}_i + y}{\bar{R}_i + \bar{B}_i + y + x^*} \\ &\geq \frac{\bar{R}_i + y^*}{\bar{R}_i + \bar{B}_i + y^* + x^*} \\ &= P(Z_{i,n+1} = 1|\{Z_j^n = b_j^n\}_{j=1}^N). \end{aligned}$$

□

Intuitively, this result states that the process is self-reinforcing; i.e., higher rates of infection increase the likelihood of the infection recurring. We will use this lemma in the proof of Lemma 3.1.5.

Consider the general form of $\tilde{I}_n$ given in (2.7). We will be comparing the performance of two curing initializations, $\mathbf{B}$ and $\mathbf{B}^*$, in limiting the spread of infection. In order to differentiate between the resultant probabilities, we will use * to denote any probability obtained using the curing initialization $\mathbf{B}^*$, e.g., $\tilde{I}_n^* = \tilde{I}_n(\mathbf{B}^*)$.

Lemma 3.1.5. *Consider a general network* $\mathcal{G} = (V, \mathcal{E})$ *equipped with the Polya network contagion model. Consider two different curing initializations* $\mathbf{B} = (B_1, \ldots, B_N)$ *and* $\mathbf{B}^* = (B_1^*, \ldots, B_N^*)$. *Suppose that* $S_{i,t}^*(a_1^t, \ldots, a_N^t) \leq S_{i,t}(a_1^t, \ldots, a_N^t)$ *for all*

$(i,t) \in V \times \{1,\dots,n\}$ and any sequence of draws $\boldsymbol{a}^n = (a_{1,s},\dots,a_{N,s})_{s=1}^n \in \{0,1\}^{N\times n}$. Then $\tilde{I}^*_{n+1} \le \tilde{I}_{n+1}$.

Proof. First, denoting $\boldsymbol{a}_s = (a_{1,s},\dots,a_{N,s})$, we can write using (2.8) that

$$\begin{aligned} P(Z_{i,n+1}=1) &= \sum_{\boldsymbol{a}^n} P(Z_{i,n+1}=1|\{Z_j^n = a_j^n\}_{j=1}^N) P(\{Z_j^n = a_j^n\}_{j=1}^N) \\ &= \sum_{\boldsymbol{a}^n} [P(Z_{i,n+1}=1|\{Z_j^n = a_j^n\}_{j=1}^N) \prod_{t=1}^n P(\{Z_{i,t} = a_{i,t}\}_{i=1}^N | \{Z_i^{t-1} = a_i^{t-1}\}_{i=1}^N]. \end{aligned}$$

Instead of using this form for $P(Z_{i,n+1}=1)$ and expanding as in (2.6), we can instead recursively apply the law of total probability to break down $P(Z_{i,n+1}=1)$ directly as follows:

$$\begin{aligned} P(Z_{i,n+1}=1) = \sum_{\boldsymbol{a}_1} [P(Z_1 = \boldsymbol{a}_1) \sum_{\boldsymbol{a}_2} [P(Z_2 = \boldsymbol{a}_2 | Z_1 = \boldsymbol{a}_1) \cdots \\ \cdots \sum_{\boldsymbol{a}_n} [P(Z_n = \boldsymbol{a}_n | Z^{n-1} = \boldsymbol{a}^{n-1}) P(Z_{i,n+1}=1 | Z^n = \boldsymbol{a}^n)]]]. \end{aligned} \tag{3.2}$$

Now that we have this equation in this form, we iteratively replace terms to obtain the desired inequality. First, consider the sum over $\boldsymbol{a}_n$. We start with the fact that $S_{i,n} = P(Z_{i,n+1}=1|Z^n)$, and we are assuming that $S^*_{i,n} \le S_{i,n}$ for any realization of the contagion process. We have

$$\begin{aligned} &\sum_{\boldsymbol{a}_n} P(Z_n = \boldsymbol{a}_n | Z^{n-1} = \boldsymbol{a}^{n-1}) P(Z_{i,n+1}=1 | Z^n = \boldsymbol{a}^n) \\ =& \sum_{\boldsymbol{a}_n} \prod_{j=1}^N (S_{j,n-1})^{a_{j,n}} (1 - S_{j,n-1})^{1-a_{j,n}} P(Z_{i,n+1}=1 | Z^n = \boldsymbol{a}^n) \\ \ge& \sum_{\boldsymbol{a}_n} \prod_{j=1}^N (S_{j,n-1})^{a_{j,n}} (1 - S_{j,n-1})^{1-a_{j,n}} P^*(Z_{i,n+1}=1 | Z^n = \boldsymbol{a}^n). \end{aligned} \tag{3.3}$$

The equality comes from the fact that the draw processes at time n are independent given the entire draw history up to that point, while the inequality arises from our assumption that $S^*_{i,n} \le S_{i,n}$.

Next, we replace $S_{j,n-1}$ with $S^*_{j,n-1}$ in (3.3) term by term; in particular, we first sum out $a_{1,n}$. Here we use the convention that $\boldsymbol{a}_n = (a_{1,n}, \boldsymbol{b}_n)$, i.e., $\boldsymbol{b}_n$ omits the first component of the vector $\boldsymbol{a}_n$. Likewise, we will use the convention that $\boldsymbol{a}^s = \{a_{1,s}, \boldsymbol{b}^s\}$. As a result, (3.3) yields

$$\begin{aligned}
&\sum_{\boldsymbol{a}_n} \prod_{j=1}^{N} (S_{j,n-1})^{a_{j,n}} (1 - S_{j,n-1})^{1-a_{i,n}} P^*(Z_{i,n+1} = 1 | Z^n = \boldsymbol{a}^n) \\
&= \sum_{\boldsymbol{b}_n} [\prod_{j=2}^{N} (S_{j,n-1})^{a_{j,n}} (1 - S_{j,n-1})^{1-a_{i,n}}] \\
&\qquad \times (S_{1,n-1} P^*(Z_{i,n+1} = 1 | Z^n = \{1, \boldsymbol{b}^n\}) + (1 - S_{1,n-1}) P^*(Z_{i,n+1} = 1 | Z^n = \{0, \boldsymbol{b}^n\})).
\end{aligned}$$

We replace $S_{1,n-1}$ in this expression with $S^*_{1,n-1}$, and then subtract it from the above to get

$$\begin{aligned}
&\sum_{\boldsymbol{b}_n} [\prod_{j=2}^{N} (S_{j,n-1})^{a_{j,n}} (1 - S_{j,n-1})^{1-a_{i,n}}] \\
&\qquad \times [(S_{1,n-1} P^*(Z_{i,n+1} = 1 | Z^n = \{1, \boldsymbol{b}^n\}) + (1 - S_{1,n-1}) P^*(Z_{i,n+1} = 1 | Z^n = \{0, \boldsymbol{b}^n\})) \\
&\qquad - (S^*_{1,n-1} P^*(Z_{i,n+1} = 1 | Z^n = \{1, \boldsymbol{b}^n\}) + (1 - S^*_{1,n-1}) P^*(Z_{i,n+1} = 1 | Z^n = \{0, \boldsymbol{b}^n\}))] \\
&= \sum_{\boldsymbol{b}_n} [\prod_{j=2}^{N} (S_{j,n-1})^{a_{j,n}} (1 - S_{j,n-1})^{1-a_{i,n}}] \\
&\qquad \times (S_{1,n-1} - S^*_{1,n-1})(P^*(Z_{i,n+1} = 1 | Z^n = \{1, \boldsymbol{b}^n\} - P^*(Z_{i,n+1} = 1 | Z^n = \{0, \boldsymbol{b}^n\}) \\
&\geq 0.
\end{aligned}$$

We note that

$$P^*(Z_{i,n+1} = 1 | Z^n = \{1, \boldsymbol{b}^n\}) \geq P^*(Z_{i,n+1} = 1 | Z^n = \{0, \boldsymbol{b}^n\})$$

from Lemma 3.1.4, and $S^*_{1,n-1} \leq S_{1,n-1}$ as per our assumption. Thus, we ultimately

obtain that

$$\begin{aligned}
&\sum_{\boldsymbol{a}_n}\prod_{j=1}^{N}(S_{j,n-1})^{a_{j,n}}(1-S_{j,n-1})^{1-a_{i,n}}P^*(Z_{i,n+1}=1|Z^n=\boldsymbol{a}^n)\\
\geq&\sum_{\boldsymbol{a}_n}[\prod_{j=2}^{N}(S_{j,n-1})^{a_{j,n}}(1-S_{j,n-1})^{1-a_{i,n}}\\
&\qquad\times(S^*_{1,n-1})^{a_{1,n}}(1-S^*_{1,n-1})^{1-a_{1,n}}P^*(Z_{i,n+1}=1|Z^n=\boldsymbol{a}^n)].
\end{aligned}$$

We follow this same process of replacing $S_{j,n}$ with $S^*_{j,n}$ for each $j \in V$ in order to conclude that

$$\begin{aligned}
&\sum_{\boldsymbol{a}_n}\prod_{j=1}^{N}(S_{j,n-1})^{a_{j,n}}(1-S_{j,n-1})^{1-a_{i,n}}P(Z_{i,n+1}=1|Z^n=\boldsymbol{a}^n)\\
\geq&\sum_{\boldsymbol{a}_n}\prod_{j=1}^{N}(S^*_{j,n-1})^{a_{j,n}}(1-S^*_{j,n-1})^{1-a_{i,n}}P^*(Z_{i,n+1}=1|Z^n=\boldsymbol{a}^n)\\
=&\sum_{\boldsymbol{a}_n}P^*(Z_n=\boldsymbol{a}_n|Z^{n-1}=\boldsymbol{a}^{n-1})P^*(Z_{i,n+1}=1|Z^n=\boldsymbol{a}^n)\\
=&P^*(Z_{i,n+1}=1|Z^{n-1}=\boldsymbol{a}^{n-1}).
\end{aligned}$$

Returning to our expression in (3.2), we have

$$\begin{aligned}
P(Z_{i,n+1}=1)=&\sum_{\boldsymbol{a}_1}[P(Z_1=\boldsymbol{a}_1)\sum_{\boldsymbol{a}_2}[P(Z_2=\boldsymbol{a}_2|Z_1=\boldsymbol{a}_1)\cdots\\
&\cdots\sum_{\boldsymbol{a}_n}[P(Z_n=a_n|Z^{n-1}=\boldsymbol{a}^{n-1})P(Z_{i,n+1}=1|Z^n=\boldsymbol{a}^n)]]]\\
\geq&\sum_{\boldsymbol{a}_1}[P(Z_1=\boldsymbol{a}_1)\sum_{\boldsymbol{a}_2}[P(Z_2=\boldsymbol{a}_2|Z_1=\boldsymbol{a}_1)\cdots\\
&\cdots\sum_{\boldsymbol{a}_{n-1}}[P(Z_{n-1}=\boldsymbol{a}_{n-1}|Z^{n-2}=\boldsymbol{a}^{n-2})P^*(Z_{i,n+1}=1|Z^{n-1}=\boldsymbol{a}^{n-1})]]].
\end{aligned}$$

We can follow the same procedure for the sum over $\boldsymbol{a}_{n-1}$ as we did for the sum over

$\boldsymbol{a}_n$ in order to find that

$$\begin{aligned}
&\sum_{\boldsymbol{a}_{n-1}}[P(Z_{n-1}=\boldsymbol{a}_{n-1}|Z^{n-2}=\boldsymbol{a}^{n-2})P^*(Z_{i,n+1}=1|Z^{n-1}=\boldsymbol{a}^{n-1})]\\
\geq&\sum_{\boldsymbol{a}_{n-1}}[P^*(Z_{n-1}=\boldsymbol{a}_{n-1}|Z^{n-2}=\boldsymbol{a}^{n-2})P^*(Z_{i,n+1}=1|Z^{n-1}=\boldsymbol{a}^{n-1})]\\
=&P^*(Z_{i,n+1}=1|Z^{n-2}=\boldsymbol{a}^{n-2}).
\end{aligned}$$

We can then substitute this back into (3.2) and repeat for each summand that appears, ultimately ending up with

$$P(Z_{i,n+1}=1)\geq P^*(Z_{i,n+1}=1).$$

This will hold for each $i\in V$, giving us that $\tilde{I}_{n+1}\geq\tilde{I}^*_{n+1}$ as desired. □

Alternative proof of Lemma 3.1.5. Let $\{Y_{i,t}:t=1,\ldots,n,i=1\ldots,N\}$ be independent random variables each uniformly distributed on the unit interval [0,1]. For all $i=1,\ldots,N$ let $Z_{i,t}$ and $Z^*_{i,t}$ be inductively defined for $t=1,\ldots,n$ by

$$Z_{i,t}=\begin{cases}1 & \text{if } Y_{i,t}\leq S_{i,t-1}(\{Z_i^{t-1}\}_{i=1}^N)\\ 0 & \text{otherwise,}\end{cases}\tag{3.4}$$

and

$$Z_{i,t}=\begin{cases}1 & \text{if } Y_{i,t}\leq S^*_{i,t-1}(\{Z_i^{*,t-1}\}_{i=1}^N)\\ 0 & \text{otherwise.}\end{cases}\tag{3.5}$$

Here we assume that $S_{i,0}\geq S^*_{i,0}$ for all $i=1\ldots,N$.

Note that since $P(Z_{i,t}=1|\{Z_i^{t-1}\}_{i=1}^N)=S_{i,t-1}(\{Z_i^{t-1}\}_{i=1}^N)$ and $P(Z^*_{i,t}=1|\{Z_i^{*,t-1}\}_{i=1}^N)=S^*_{i,t-1}(\{Z_i^{*,t-1}\}_{i=1}^N)$, for all $i=1,\ldots,N$ and $t=1,\ldots,n$, the processes have the required Polya network process distribution with initializations $\mathbf{B}$ and $\mathbf{B}^*$, respectively.

Claim 3.1.6. *Suppose for some $t\in\{1,\ldots,n-1\}$ the sequence $\{Z_i^{t-1}\}_{i=1}^N$ dominates*

$\{Z_i^{*,t-1}\}_{i=1}^N$ *in the sense that* $Z_{i,s} \geq Z_{i,s}^*$ *for all* $i = 1, \ldots, N$ *and* $s = 1, \ldots, t-1$. *Then* $\{Z_i^t\}_{i=1}^N$ *dominates* $\{Z_i^{*,t}\}_{i=1}^N$.

Proof of Claim. Repeated application of Lemma 3.1.4 shows that $S_{i,t-1}(\{Z_i^{t-1}\}_{i=1}^N) \geq S_{i,t-1}^*(\{Z_i^{*,t-1}\}_{i=1}^N)$ for all $i = 1, \ldots, N$. (Specifically, since $Z_{i,s} \geq Z_{i,s}^*$ means that either $Z_{i,s} = Z_{i,s}^*$ or $Z_{i,s} = 1$ and $Z_{i,s}^* = 0$, Lemma 3.1.4 is applied $K = |\{(i,s) : Z_{i,s} \neq Z_{i,s}^*\}|$ times.) This, (3.4), and (3.5) then give $Z_{i,t} \geq Z_{i,t}^*$, which in turn implies that $\{Z_i^t\}_{i=1}^N$ dominates $\{Z_i^{*,t}\}_{i=1}^N$. □

Using Claim 3.1.6 for $t = 0$, which vacuously holds by the assumption that $S_{i,0} \geq S_{i,0}^*$ for all $i = 1, \ldots, N$, we obtain $Z_{i,1} \geq Z_{i,1}^*$, $i = 1, \ldots, N$. Applying Claim 3.1.6 inductively for $t = 2, \ldots, n$, we obtain that

$$Z_{i,t} \geq Z_{i,t}^*, \quad i = 1, \ldots, N, \quad t = 1, \ldots, n$$

with probability one. Thus the event $\{Z_{i,t}^* = 1\}$ implies $\{Z_{i,t} = 1\}$ and so $P(Z_{i,t}^* = 1) \leq P(Z_{i,t} = 1)$ for all $i = 1, \ldots, N$ and $t = 1, \ldots, n$. Therefore

$$\tilde{I}_t^* \leq \tilde{I}_t^*$$

for all $t = 1, \ldots, n$ as claimed. □

We can now use this result in order to extend Lemma 3.1.3 to the case for a general time n.

Proposition 3.1.7. *Consider a general network* $\mathcal{G} = (V, \mathcal{E})$ *equipped with the Polya network contagion model. If for any nodes* $i, j \in V$ *we have* $\mathcal{N}_i' \subset \mathcal{N}_j'$, *then it can be assumed that* $B_i = 0$ *when minimizing* $\tilde{I}_n$ *over the curing initialization. Likewise, it can be assumed that* $R_i = 0$ *when maximizing* $\tilde{I}_n$ *over the infection initialization.*

Proof. Without loss of generality assume $\mathcal{N}_1' \subset \mathcal{N}_2'$. Consider the general form of $S_{i,n}(a_1^n, \ldots, a_N^n)$:

$$\frac{\bar{R}_i + \sum_{j \in \mathcal{N}_i'} \sum_{t=1}^{n} a_{j,t} \Delta_{r,j}(t)}{\bar{R}_i + \bar{B}_i + \sum_{j \in \mathcal{N}_i'} \sum_{t=1}^{n} (a_{j,t} \Delta_{r,j}(t) + (1 - a_{j,t}) \Delta_{b,j}(t))}. \tag{3.6}$$

Letting $B_1^* = 0$, and $B_2^* = B_1 + B_2$, as in the proof of Lemma 3.1.3, we have for any $i \in V$ that $\bar{B}_i^* \geq \bar{B}_i$. Plugging into (3.6) yields that $S_{i,n}^* \leq S_{i,n}$ for any realization of the draw process. This will hold for any $i \in V$ and for any time t. We apply Lemma 3.1.5 to get the desired result. □

Essentially what we have proven here is that an optimal solution for Problem 3.1.1 will allocate no resources to outer nodes. Though this result may seem somewhat intuitive, because of the complex nature of $\tilde{I}_n$ the proof as demonstrated is rather involved. Figure 3.1 shows an example of how curing resources would be distributed for solutions to Problem 3.1.1 in accordance with Proposition 3.1.7.

3.1.2 Symmetry

The second simplification we make regarding solutions to Problem 3.1.1 has to do with networks that possess symmetry properties in some sense. First we need to define what is meant by this, and so we introduce the following concepts, which are described in detail in [22, 7, 14], along with other graph theory concepts.

Definition 3.1.8. *An automorphism of a graph $\mathcal{G} = (V, \mathcal{E})$ is a permutation σ of V, such that $(u, v) \in \mathcal{E}$ iff $(\sigma(u), \sigma(v)) \in \mathcal{E}$. The set of automorphisms of a given graph forms a group under the composition operation, called the automorphism group of $\mathcal{G}$ and denoted $Aut(\mathcal{G})$.*

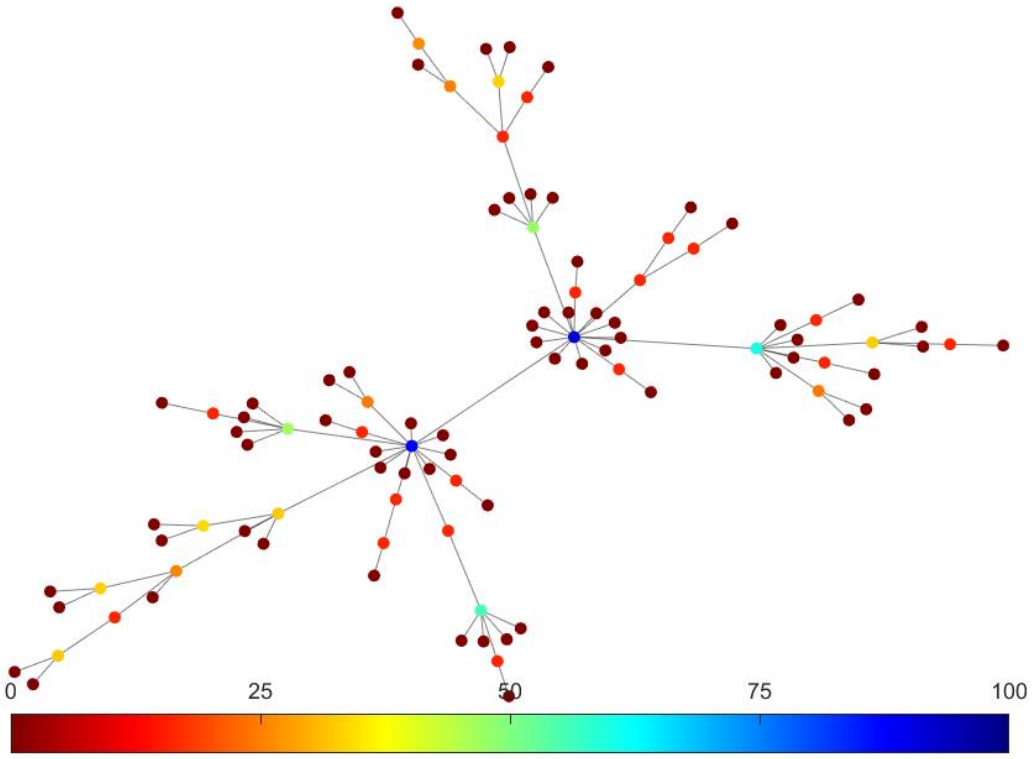
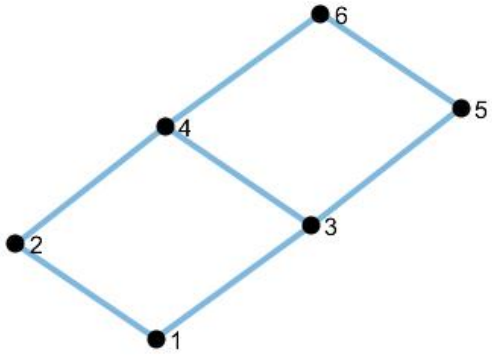


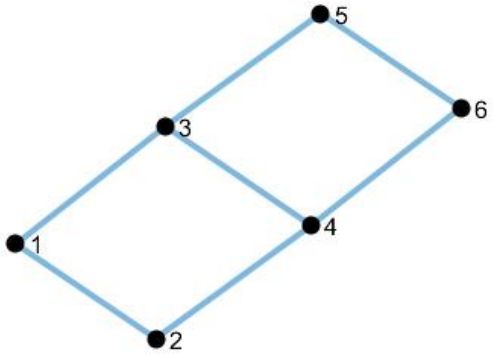
Figure 3.1: Sample distribution of curing initialization parameters for a 100 node Barabasi-Albert network, obtained using gradient descent (see Proposition 3.2.1 and Algorithm 1). The colour scale indicates how many resources are allocated to each node. An initialization budget of $\mathcal{B}_b = 1000$ is used for this example, with an infection initialization of $R_i = 100$, $\forall i \in V$.

Definition 3.1.9. *For a graph $\mathcal{G} = (V, \mathcal{E})$, if $H \leq Aut(\mathcal{G})$ is a subgroup of automorphisms of $\mathcal{G}$, then $u, v \in V$ are similar under H if there exists an automorphism in H which maps u to v. Equivalence classes defined by similarity under H are called orbits of the graph $\mathcal{G}$ by H. The sets of orbits by H form a partition of the vertices of $\mathcal{G}$, called an orbit partition.*

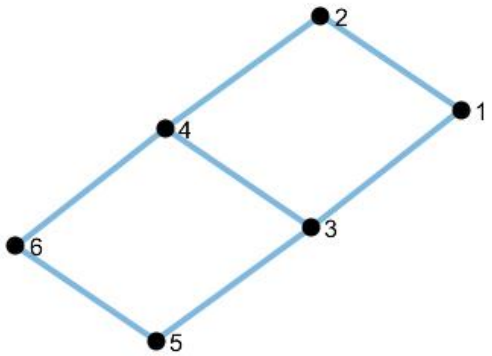
An automorphism is an isomorphism from $\mathcal{G}$ to itself. If there exists an automorphism which maps between two different nodes, it means that those nodes are



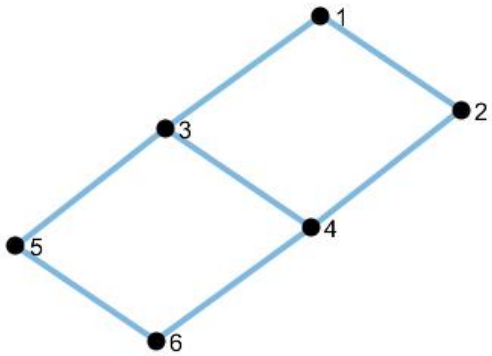
(a) Sample labelling scheme 1.



(b) Sample labelling scheme 2.



(c) Sample labelling scheme 3.



(d) Sample labelling scheme 4.

Figure 3.2: A sample network and 4 labelling schemes that give the same adjacency matrix. Because the edge sets are preserved, an automorphism can be used to map each node within a set of labels to its respective couterpart in any of the alternative labelling schemes depicted.

distinguishable only by their labels. This is illustrated by Figure 3.2. For our purposes, we consider "symmetry" between nodes characterized by automorphisms. An orbit partition is a way to divide a network based on these symmetries. For example, if we consider the network in Figure 3.2, the orbit partition of $\mathcal{G}$ by $Aut(\mathcal{G})$ is given by $P = \{\{1, 2, 5, 6\}, \{3, 4\}\}$.

We want to prove, that under certain conditions, sets of nodes that are similar

under some $H \leq Aut(\mathcal{G})$ will be treated equivalently; i.e., they will all be given the same number of resources by an optimal policy. We start with a preliminary result.

Lemma 3.1.10. *Consider a general network $\mathcal{G} = (V, \mathcal{E})$ with a given initialization $(\mathbf{B}, \mathbf{R}) = (B_1, \ldots, B_N, R_1, \ldots, R_N)$. Suppose there exists a subset of nodes $V' \subset V$ such that $\bar{R}_i = \bar{R}_j$ for all $i, j \in V'$. Now suppose we can find some alternate curing initialization $\mathbf{B}^* = (B_1^*, \ldots, B_N^*)$ such that $\bar{B}_i^* = \frac{1}{|V'|}\sum_{j\in V'} \bar{B}_j$ for all $i \in V'$. Then $\sum_{i\in V'} P^*(Z_{i,1} = 1) \leq \sum_{i\in V'} P(Z_{i,1} = 1)$.*

Proof. We prove this result by induction on $|V'|$. Start with the simplest case, where $|V'| = 2$. Consider two nodes $i, j \in V$ such that $\bar{R}_i = \bar{R}_j$. Let $\bar{B}_i^* = \bar{B}_j^* = \frac{1}{2}(\bar{B}_i + \bar{B}_j)$. We have

$$\begin{aligned} P(Z_{i,1} = 1) + P(Z_{j,1} = 1) &= \frac{\bar{R}_i}{\bar{R}_i + \bar{B}_i} + \frac{\bar{R}_j}{\bar{R}_j + \bar{B}_j} \\ &= \frac{\bar{R}_i}{\bar{R}_i + \bar{B}_i} + \frac{\bar{R}_i}{\bar{R}_i + \bar{B}_j}, \qquad (3.7) \\ P^*(Z_{i,1} = 1) + P^*(Z_{j,1} = 1) &= \frac{\bar{R}_i}{\bar{R}_i + \bar{B}_i^*} + \frac{\bar{R}_j}{\bar{R}_j + \bar{B}_j^*} \\ &= \frac{2\bar{R}_i}{\bar{R}_i + \frac{1}{2}(\bar{B}_i + \bar{B}_j)}. \qquad (3.8) \end{aligned}$$

We want to show (3.7) is no less than (3.8). Taking the difference we get

$$\begin{aligned} &\frac{\bar{R}_i}{\bar{R}_i + \bar{B}_i} + \frac{\bar{R}_i}{\bar{R}_i + \bar{B}_j} - \frac{2\bar{R}_i}{\bar{R}_i + \frac{1}{2}(\bar{B}_i + \bar{B}_j)} \\ =&\bar{R}_i \left[\frac{2\bar{R}_i + \bar{B}_i + \bar{B}_j}{(\bar{R}_i + \bar{B}_i)(\bar{R}_i + \bar{B}_j)} - \frac{2}{\bar{R}_i + \frac{1}{2}(\bar{B}_i + \bar{B}_j)}\right] \\ =&\bar{R}_i \left[\frac{(2\bar{R}_i + \bar{B}_i + \bar{B}_j)(\bar{R}_i + \frac{1}{2}(\bar{B}_i + \bar{B}_j)) - 2(\bar{R}_i + \bar{B}_i)(\bar{R}_i + \bar{B}_j)}{(\bar{R}_i + \bar{B}_i)(\bar{R}_i + \bar{B}_j)(\bar{R}_i + \frac{1}{2}(\bar{B}_i + \bar{B}_j))}\right] \\ =&\bar{R}_i \left[\frac{2\bar{R}_i^2 + 2\bar{R}_i(\bar{B}_i + \bar{B}_j) + \frac{1}{2}(\bar{B}_i + \bar{B}_j)^2 - 2\bar{R}_i^2 - 2\bar{R}_i(\bar{B}_i + \bar{B}_j) - 2\bar{B}_i\bar{B}_j}{(\bar{R}_i + \bar{B}_i)(\bar{R}_i + \bar{B}_j)(\bar{R}_i + \frac{1}{2}(\bar{B}_i + \bar{B}_j))}\right] \\ =&\bar{R}_i \left[\frac{\frac{1}{2}(\bar{B}_i^2 + \bar{B}_j^2 - 2\bar{B}_i\bar{B}_j)}{(\bar{R}_i + \bar{B}_i)(\bar{R}_i + \bar{B}_j)(\bar{R}_i + \frac{1}{2}(\bar{B}_i + \bar{B}_j))}\right] \end{aligned}$$

$$=\bar{R}_i\left[\frac{\frac{1}{2}(\bar{B}_i-\bar{B}_j)^2}{(\bar{R}_i+\bar{B}_i)(\bar{R}_i+\bar{B}_j)(\bar{R}_i+\frac{1}{2}(\bar{B}_i+\bar{B}_j))}\right]$$

$$\geq 0.$$

Now, consider a subset of nodes $V' \subset V$ such that $\bar{R}_i = \bar{R}_j$ for all $i, j \in V'$, and suppose $|V'| = n$. Without loss of generality, let $V' = \{1, \ldots, n\}$. We assume the result holds for any subset of $n-1$ of these nodes. Thus, we can write that

$$\begin{aligned}\sum_{i=1}^{n-1} P(Z_{i,1}=1) &= \sum_{i=1}^{n-1} \frac{\bar{R}_i}{\bar{R}_i+\bar{B}_i} \\ &\geq (n-1)\frac{\bar{R}_1}{\bar{R}_1+\frac{1}{n-1}\sum_{i=1}^{n-1}\bar{B}_i}.\end{aligned}$$

Let $\bar{B} = \frac{1}{n-1}\sum_{i=1}^{n-1}\bar{B}_i$, and let $\bar{B}_i^* = \frac{1}{n}((n-1)\bar{B}+\bar{B}_n)$, $\forall i \in V'$. We note that $\bar{B}_i^* = \frac{1}{n}\sum_{j\in V'}\bar{B}_j$ for all $i \in V'$. We have

$$\sum_{i\in V'} P(Z_{i,1}=1) \geq (n-1)\frac{\bar{R}_1}{\bar{R}_1+\bar{B}} + \frac{\bar{R}_1}{\bar{R}_1+\bar{B}_n}, \tag{3.9}$$

$$\sum_{i\in V'} P^*(Z_{i,1}=1) = n\frac{\bar{R}_1}{\bar{R}_1+\frac{1}{n}((n-1)\bar{B}+\bar{B}_n)}. \tag{3.10}$$

Now to prove the result we need only to show that the right-hand side of (3.9) is no smaller than (3.10). Taking the difference we get

$$\begin{aligned}&\bar{R}_1\left[\frac{(n-1)(\bar{R}_1+\bar{B}_n)+(\bar{R}_1+\bar{B})}{(\bar{R}_1+\bar{B})(\bar{R}_1+\bar{B}_n)} - \frac{n}{\bar{R}_1+\frac{1}{n}((n-1)\bar{B}+\bar{B}_n)}\right] \\ =&\bar{R}_1\left[\frac{n\bar{R}_1+(n-1)\bar{B}_n+\bar{B}}{(\bar{R}_1+\bar{B})(\bar{R}_1+\bar{B}_n)} - \frac{n}{\bar{R}_1+\frac{1}{n}((n-1)\bar{B}+\bar{B}_n)}\right] \\ =&\bar{R}_1\left[\frac{(n\bar{R}_1+(n-1)\bar{B}_n+\bar{B})(\bar{R}_1+\frac{1}{n}((n-1)\bar{B}+\bar{B}_n)-n(\bar{R}_1+\bar{B})(\bar{R}_1+\bar{B}_n)}{(\bar{R}_1+\bar{B})(\bar{R}_1+\bar{B}_n)(\bar{R}_1+\frac{1}{n}((n-1)\bar{B}+\bar{B}_n))}\right] \\ =&\bar{R}_1\left[\frac{\frac{n-1}{n}(\bar{B}^2+\bar{B}_n^2-2\bar{B}\bar{B}_n)}{(\bar{R}_1+\bar{B})(\bar{R}_1+\bar{B}_n)(\bar{R}_1+\frac{1}{n}((n-1)\bar{B}+\bar{B}_n))}\right] \\ =&\bar{R}_1\left[\frac{\frac{n-1}{n}(\bar{B}-\bar{B}_n)^2}{(\bar{R}_1+\bar{B})(\bar{R}_1+\bar{B}_n)(\bar{R}_1+\frac{1}{n}((n-1)\bar{B}+\bar{B}_n))}\right] \\ \geq&0.\end{aligned}$$

□

We can now use this result to help prove that for networks with symmetry, solutions to Problem 3.1.1 will be symmetric in nature as well. Suppose for a graph $\mathcal{G} = (V, \mathcal{E})$ we have an automorphism σ of $\mathcal{G}$. Assuming $\mathcal{G}$ is finite, we have that $Aut(\mathcal{G})$ is a finite group, and thus we can use σ to generate a cyclic subgroup $\langle\sigma\rangle$ of $Aut(\mathcal{G})$. The subgroup $\langle\sigma\rangle$ is necessarily finite and of finite order. We also have that for any node i in V there exists some integer $k \in \mathbb{Z}_{>0}$ such that $i = \sigma^k(i)$. We will denote by k_i the smallest such integer for which this holds for a given $i \in V$.

Proposition 3.1.11. *Consider a general network $\mathcal{G} = (V, \mathcal{E})$ with a given initialization $(\mathbf{B}, \mathbf{R})$ and consider an automorphism σ of $\mathcal{G}$. Let m be the order of the cyclic subgroup $\langle\sigma\rangle$ of $Aut(\mathcal{G})$. Suppose $R_i = R_{\sigma(i)}$, for all $i \in V$. Let $\mathbf{B}^*$ be an alternate curing initialization such that $B_i^* = \frac{1}{m}\sum_{j=1}^{m} B_{\sigma^j(i)}$ for all $i \in V$. Then, $\tilde{I}_1^* \leq \tilde{I}_1$.*

Proof. Note that for a given $i \in V$, $\bar{R}_i = \sum_{j\in\mathcal{N}_i'} R_j$. We have that if $j \in \mathcal{N}_i'$ then $\sigma^k(j) \in \mathcal{N}_{\sigma^k(i)}'$ for any $k \in \mathbb{Z}_{>0}$. Since by assumption $R_j = R_{\sigma(j)} = R_{\sigma^k(j)}$ for all $j \in \mathcal{N}_i'$, we have that $\bar{R}_i = \bar{R}_{\sigma^k(i)}$ for all $i \in V$ and any positive integer k. From Lemma 3.1.10, we need only to prove that $\bar{B}_i^* = \frac{1}{k_i}\sum_{j=1}^{k_i} \bar{B}_{\sigma^j(i)}$, for all $i \in V$. We start with

$$\bar{B}_i^* = \sum_{j\in\mathcal{N}_i'} B_j^* = \sum_{j\in\mathcal{N}_i'} \frac{1}{m}\sum_{l=1}^{m} B_{\sigma^l(j)}.$$

We note that for a given integer p,

$$p\sum_{l=1}^{k_j} B_{\sigma^l(j)} = \sum_{l=1}^{k_j} pB_{\sigma^l(j)} = \sum_{l=1}^{k_j}(B_{\sigma^l(j)} + B_{\sigma^{l+k_j}(j)} + \cdots + B_{\sigma^{l+(p-1)k_j}(j)}) = \sum_{l=1}^{pk_j} B_{\sigma^l(j)},$$

which follows from the definition of k_j. Since m is the order of $\langle\sigma\rangle$, we have $\sigma^m(j) = j$, and since k_j is the smallest such integer for which this holds, we must have $k_j | m$ (i.e., k_j divides m). Choosing $p = \frac{m}{k_j}$ we get (using the identity between the left-most and

right-most terms in the above equation) that

$$\frac{1}{k_j}\sum_{l=1}^{k_j} B_{\sigma^l(j)} = \frac{p}{pk_j}\sum_{l=1}^{k_j} B_{\sigma^l(j)} = \frac{1}{pk_j}\sum_{l=1}^{pk_j} B_{\sigma^l(j)} = \frac{1}{m}\sum_{l=1}^{m} B_{\sigma^l(j)},$$

and thus

$$\begin{aligned}
\bar{B}_i^* =& \sum_{j\in\mathcal{N}_i'} \frac{1}{m}\sum_{l=1}^{m} B_{\sigma^l(j)} \\
=& \frac{1}{m}\sum_{l=1}^{m}\sum_{j\in\mathcal{N}_i'} B_{\sigma^l(j)} \\
=& \frac{1}{m}\sum_{l=1}^{m}\sum_{j\in\mathcal{N}'_{\sigma^l(i)}} B_j \\
=& \frac{1}{m}\sum_{l=1}^{m} \bar{B}_{\sigma^l(i)} \\
=& \frac{1}{k_i}\sum_{l=1}^{k_i} \bar{B}_{\sigma^l(i)}.
\end{aligned}$$

The result then follows from Lemma 3.1.10. □

Another way to think about Proposition 3.1.11 is that the optimal allocation distributes resources equally within sets of the orbit partition by $\langle\sigma\rangle$. We assume that infection resources are distributed evenly within this orbit partition, a consequence of assuming that $R_i = R_{\sigma(i)}$ for all $i \in V$. We then prove redistributing curing resources within each set of the partition, i.e., letting $B_i^* = \frac{1}{m}\sum_{j=1}^{m} B_{\sigma^j(i)}$ for all $i \in V$, will result in a decrease in the overall average infection rate.

Due to the complicated nature of $\tilde{I}_n$, it is difficult to analytically prove this result for time $n > 1$. Also, while Proposition 3.1.11 can be applied to a given automorphism, we want to push this slightly further. Consider a network $\mathcal{G} = (V, \mathcal{E})$, and the orbit partition of $\mathcal{G}$ by $Aut(\mathcal{G})$, denoted $P = \{V_1, \ldots, V_q\}$, where $q \leq N$

For any given initialization, $(\mathbf{B}, \mathbf{R})$, for which $\bar{R}_i = \bar{R}_j$ if i and j are similar under

$Aut(\mathcal{G})$, we postulate that we can reduce the average infection rate $\tilde{I}_n$ by redistributing curing resources evenly within each set of the partition. To elaborate, for each $i \in V$, if $i \in V_k$ we conjecture that taking $B_i^* = \frac{1}{|V_k|} \sum_{j \in V_k} B_j$ yields that $\tilde{I}_n(\mathbf{B}^*) \leq \tilde{I}_n(\mathbf{B})$, for any n.

We provide some empirical evidence to support this claim. We again refer to the simple 6 node network provided in Figure 3.2, remembering the orbit partition by $Aut(\mathcal{G})$ for this $\mathcal{G}$ is given by $P = \{\{1, 2, 5, 6\}, \{3, 4\}\}$. In the first case we use a uniform initial distribution of red balls, $\mathbf{R}$, and a random allocation of black balls, $\mathbf{B}$. We then define an alternate curing initialization, $\mathbf{B}^*$, that redistributes resources evenly within each set of the orbit partition by $Aut(\mathcal{G})$, as described previously. We perform the draw process, averaging the draw values over a series of trials to obtain a measure for the empirical average infection rate. The results of this simulation can be found in Figure 3.3.

We see a clear reduction in the empirical average infection rate as a result of redistributing resources evenly within sets of the orbit partition, and this holds regardless of n. This result suggests that any optimal policy for Problem 3.1.1 will distribute resources evenly among nodes that are similar under some subgroup of $Aut(\mathcal{G})$, under equivalent conditions (i.e., when the infection resources are likewise uniformly allocated among similar nodes). This knowledge is useful in verifying what policies may be optimal. Additionally, this information about the structure of the optimal allocation is useful when developing heuristics.

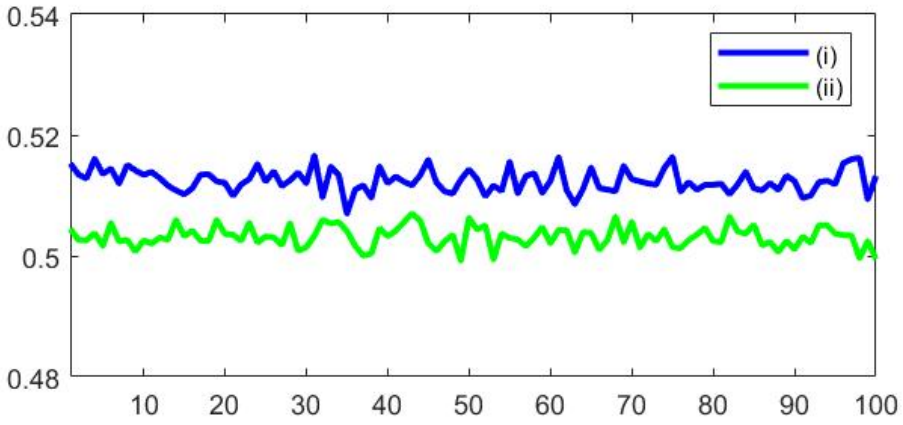


Figure 3.3: Plot of average infection rate versus time that demonstrates improved performance as a result of using uniform allocation of resources within sets of the orbit partition under $Aut(\mathcal{G})$, for simple network in Figure 3.2. In each case, 1000 trials were run, and the resulting plots show the empirical average infection rate at each time n. An infection initialization of $R_i = 10$, $\forall i \in V$ was used, with a fixed $\Delta_r = \Delta_b = 1$. The trials in (i) randomly distributed curing resources, $\mathbf{B}$, from a fixed budget $\mathcal{B}_b = 60$, while for (ii) these randomly allocated resources were redistributed uniformly within the sets of the orbit partition.

3.2 Gradient Descent

As done in [18] and further explored in Chapter 4, we implement a gradient descent algorithm [5] to find solutions to Problem 3.1.1. We first verify that this strategy can legitimately be employed for our initialization cases.

For the curing problem presented in [18], the objective was to minimize the limiting average infection rate. As this measure is rather complicated to analyze, a proxy measure given by the expected average network exposure, $E[\tilde{S}_n|\mathcal{F}_{n-1}]$, is used instead. For the initialization problem the equivalent method would be to study this value for time $n = 1$, at which point this simply reduces to minimizing the average infection rate at time one, given by $\tilde{I}_1$, as a function of the curing initialization $\mathbf{B}$.

In Problem 3.1.1, we assign curing resources subject to a budget $\mathcal{B}_b$. If for a

general time n we instead use $\tilde{I}_1$ as a simple proxy, it is easy to use gradient flow techniques to find an optimal policy for this one-step problem. We note that any optimal policy will make full use of the budget $\mathcal{B}_b$.

Proposition 3.2.1. (Gradient descent conditions): *Consider a general network $\mathcal{G} = (V, \mathcal{E})$ equipped with the Polya network contagion model. Suppose for a given infection initialization, $\mathbf{R}$, we have $\bar{R}_i > 0$ for each $i \in V$. Then the average infection rate at time one, $\tilde{I}_1$, is convex with respect to the curing initialization parameters, $\mathbf{B}$. Moreover, the feasible set*

$$\mathcal{X} = \left\{ \{B_i\}_{i=1}^N \in \mathbb{R}_{\geq 0}^N \middle| \sum_{i=1}^N B_i = \mathcal{B}_b \right\}$$

is convex and compact.

The proof of this proposition is rather straightforward, with the matter of convexity of $\tilde{I}_1$ having been discussed earlier in this chapter. Of note, however is the fact that we require $\bar{R}_i > 0$ for each $i \in V$, in order to ensure our feasible set $\mathcal{X}$ remains valid. If this is satisfied, we can then use a constrained gradient descent method [5, Chapter 2], as described in Algorithm 1. This algorithm will converge to an optimal solution for the one-step optimization problem, but not necessarily for the n-step case.

We note that minimizing $\tilde{I}_1$ is equivalent to minimizing the average infection rate of Network Models I and II that were presented in Chapter 2. Both of these models represent the contagion process of each individual node using a classical Polya process, and in each case the $\rho_{c,i}$ is taken to be $\frac{\bar{R}_i}{\bar{T}_i}$ for each node i in the network. Since we know that the classical Polya process is stationary, the expected draw value for a given node will not change with time. Minimizing the average infection rate for these models for a general time n is thus the same as minimizing the average infection rate

Algorithm 1 Constrained gradient descent on a simplex [5]

$f(x_1, \ldots, x_N) \leftarrow$ function to be minimized

Start at an arbitrary node:
$y_1 = (y_{1,1}, \ldots, y_{1,N}) = (\mathcal{B}_b, 0, \ldots, 0)$

for $k = 1 : stoptime$ **do**
Find the direction of steepest descent:
$i = \arg\min_{j \in V} \frac{\partial f}{\partial x_j}|_{y_k}$

Move only in that direction:
$\bar{y}_{k,i} = \mathcal{B}_b$, and $\bar{y}_{k,j} = 0$ for all $j \neq i$

Select the step size using the limit minimization rule:
$\alpha_k = \arg\min_{\alpha \in [0,1]} f(y_k + \alpha(\bar{y}_k - y_k))$

Perform the gradient descent:
$y_{k+1} = y_k + \alpha_k(\bar{y}_k - y_k)$

at time $n = 1$. Thus, in the context of Problem 3.1.1, this is equivalent to minimizing the average value of $\rho_{c,i}$ within the network over our choice of $\mathbf{B}$, or of minimizing

$$\frac{1}{N}\sum_{i=1}^{N} \rho_{c,i} = \frac{1}{N}\sum_{i=1}^{N} \frac{\bar{R}_i}{\bar{T}_i} = \tilde{I}_1.$$

3.3 Single Node Curing

Consider a situation where we aim to either control the spread of infection within a network, or wish to inject an infection within the network and allow it to propogate. Suppose, however, there is a limitation in terms of how many nodes may be accessed at once (for example, advertising campaigns may be limited in scope). We wish to determine the priority of nodes within a network when only a single node may be accessed. This may also provide insight when considering general curing policies without such limitations.

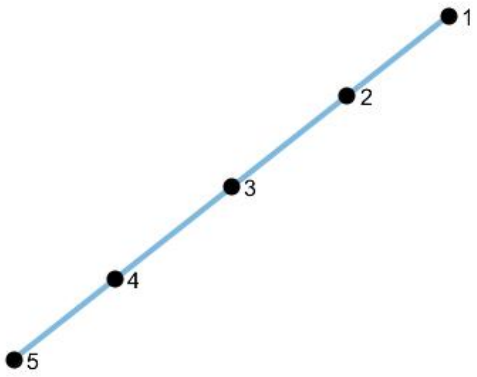


Figure 3.4: Five node path graph.

We consider a simple path graph as depicted in Figure 3.4. If we were asked which node we should target in the context of Problem 3.1.1, assuming that we could only target one node, the intuitive answer would probably be to target node 3. This node has the most neighbours and is the most centralized node, and so it takes the fewest number of steps for it to impact the rest of the network. We know from Proposition 3.1.7 that nodes 1 and 5 will be ignored by any optimal policy, and thus we only have three remaining nodes to consider. Futhermore, because of the symmetry of the network, nodes 2 and 4 are indistinguishable, further reducing this problem.

We consider a uniform infection initialization such that $R_1 = \cdots = R_5 = R$, and we take $n = 1$. If we let $\mathbf{B} = (0, B, 0, 0, 0)$, then $\bar{B}_1 = \bar{B}_2 = \bar{B}_3 = B$ and $\bar{B}_4 = \bar{B}_5 = 0$. If instead we use $\mathbf{B}^* = (0, 0, B, 0, 0)$, then $\bar{B}_2^* = \bar{B}_3^* = \bar{B}_4^* = B$ and $\bar{B}_1^* = \bar{B}_5^* = 0$. We have

$$\begin{aligned}\tilde{I}_1 &= \frac{1}{5}\sum_{i=1}^{5}\frac{\bar{R}_i}{\bar{R}_i + \bar{B}_i} \\ &= \frac{1}{5}\left(\frac{2R}{2R+\bar{B}_1} + \frac{3R}{3R+\bar{B}_2} + \frac{3R}{3R+\bar{B}_3} + \frac{3R}{3R+\bar{B}_4} + \frac{2R}{2R+\bar{B}_5}\right)\end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{5}\left(\frac{R}{R+\frac{1}{2}B} + 2\frac{R}{R+\frac{1}{3}B} + 2\right) \\
&\leq \frac{1}{5}\left(3\frac{R}{R+\frac{1}{3}B} + 2\right) \\
&= \frac{1}{5}\left(\frac{2R}{2R+\bar{B}_1^*} + \frac{3R}{3R+\bar{B}_2^*} + \frac{3R}{3R+\bar{B}_3^*} + \frac{3R}{3R+\bar{B}_4^*} + \frac{2R}{2R+\bar{B}_5^*}\right) \\
&= \tilde{I}_1^*.
\end{aligned}
$$

We are able to reduce the average infection rate at time $n = 1$ by using the curing policy $\mathbf{B}$ instead of $\mathbf{B}^*$, somewhat counter to what our intuition may tell us. This result comes because while nodes 2 and 3 have the same number of neighbours, node 2 has a higher influence over the behaviour of node 1 than node 3 has over nodes 2 or 4. Thus, when provided with a uniform red ball initialization, we find that the optimal target is node 2 (or equivalently 4), when minimizing for $\tilde{I}_1$ for the five node path graph.

If we examine the behaviour of $\tilde{I}_n$ for $n > 1$, under the same configurations, we will not come to the same conclusion. Consider Figure 3.5, which depicts the empirical average infection rate measured as a result of targeting only a single node for the 5 node path graph in Figure 3.4. As we can see, in accordance with our earlier analysis, targeting node 2 minimizes the average infection rate initially, but not as n increases. Ultimately, targeting more central nodes proves more beneficial when minimizing $\tilde{I}_n$ as $n \to \infty$.

We can extend this same reasoning to our unrestricted curing policies. We note that an optimal solution for time $n = 1$ is not necessarily optimal for all n. This means the gradient descent algorithm detailed in Algorithm 1, which converges to the optimal solution for time $n = 1$, most likely converges to only a suboptimal solution for any other n. Likewise, prioritizing nodes based on centrality will not

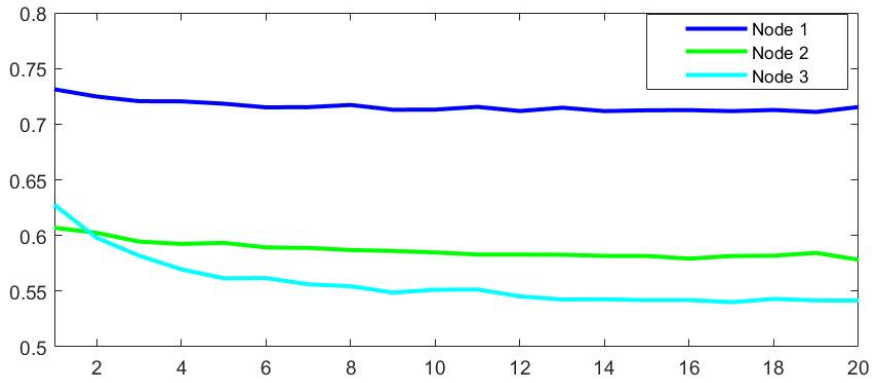


Figure 3.5: Empirical average infection rate, $\tilde{I}_n$, versus time, n, for single node curing of 5 node path graph with uniform infection initialization ($R_i = 10$ for all $i \in V$). In each case, a single node was initialized with 50 black balls, and the network Polya process was then run with fixed $\Delta_r = \Delta_b = 5$.

necessarily be optimal in the context of Problem 3.1.1 for small values of n. We wish to consider these factors when developing heuristic policies for such initialization problems in order to develop policies that are close to, or possibly more effective, than those generated using gradient descent algorithms.

3.4 Heuristic Strategies

Introduced in [16], [19] were a number of heuristic models for network curing that can be similarly applied to the initialization problems presented in this chapter. We plan here, however, to provide a heuristic model based on the analysis we have performed thus far, while also integrating some of the previously tested techniques. Since we know any optimal policy will be in accordance with Proposition 3.1.7, we can use this as our starting point.

3.4.1 Interior Node Targeting

The first strategy we consider is rather straightforward. Given a network, we determine which nodes are inner nodes, i.e., which nodes have a neighbourhood that is not a strict subset of another node's neighbourhood. We then distribute initialization resources uniformly among these nodes. If we let $\mathcal{B}_b$ represent our curing initialization budget, and let $V' = \{i \in V | \mathcal{N}'_i \not\subset \mathcal{N}'_j, \forall j \in V\}$, we then allocate our resources according to the following scheme:

$$B_i = \begin{cases} \frac{1}{|V'|}\mathcal{B}_b & \text{if } i \in V' \\ 0 & \text{else.} \end{cases}$$

We can also combine this scheme with the heuristic strategies in [16], [19]. Among these strategies, the best performing heuristic strategy allocates resources based on a combination of centrality measures, node degree and closeness centrality, as well as the infection level of each node's superurn. Since we are dealing with initialization policies, it makes little sense to do our distribution based on our infection levels; instead we implement a strategy based upon the node degree and closeness centrality.

The degree of a node is simply measured as the size of a node's neighbourhood, $|\mathcal{N}_i|$ (we can similarly use $|\mathcal{N}'_i|$, if desired). The closeness centrality of a node, C_i, is a measure used to determine a node's topological position within a network, defined as

$$C_i = \frac{1}{\sum_{j \in V} d(i,j)}, \tag{3.11}$$

where $d(i,j)$ is the shortest path length from node i to node j. Thus the node with the highest closeness centrality has the shortest average path length to every node

within the network. We can distribute resources as follows:

$$B_i = \begin{cases} \frac{|\mathcal{N}_i|C_i}{\sum_{j\in V'}|\mathcal{N}_j|C_j}\mathcal{B}_b & \text{if } i \in V' \\ 0 & \text{else.} \end{cases}$$

Note we could alternatively apply a similar strategy using "betweenness centrality" as in [19] instead.

3.4.2 Minimized Node Targeting

We can get greedier with our initialization policies, based on the idea of minimizing the number of nodes we target at once. We introduce a method that attempts to target nodes in such a manner that all nodes have resources within their immediate neighbourhood $\mathcal{N}_i'$ (i.e., ensuring $\bar{B}_i > 0$, $\forall i \in V$), while targeting the fewest possible nodes. We use Algorithm 2 to determine which nodes will be allocated resources.

This algorithm starts by locating outer nodes in the network and then targets nodes directly adjacent these outer nodes. All nodes that are within the neighbourhood of the targeted nodes are then removed from the set of nodes being evaluated. Within this new, smaller set of nodes, we once again locate outer nodes and repeat the process. In the case that there are no outer nodes within the set of nodes being evaluated, all remaining nodes are added to the set of targeted nodes.

The purpose of this algorithm is to effectively employ Proposition 3.1.7, and ensure our curing policy does not target outer nodes, while still ensuring $\bar{B}_i > 0$ for these nodes. Once the set of outer nodes is excluded from directly receiving resources, the algorithm then works inwards, attempting to minimize the number of targeted nodes.

Once the set of nodes to be targeted, V', has been determined we can once again distribute resources in one of two manners as in the previous strategy set. We can

Algorithm 2 Node Targeting Algorithm

$V \leftarrow$ set of all nodes
$V' \leftarrow$ target set
$V_{test} \leftarrow$ nodes to test for this loop

Start with $V' = \emptyset$, $V_{test} = V$
while $V_{test} \neq \emptyset$ **do**
  Identify outer nodes:
  Set $V_{outer} = \{i \in V_{test} | \mathcal{N}_i' \subset \mathcal{N}_j', \text{ for some } j \in V_{test}\}$
  if $V_{outer} = \emptyset$ **then**
    $V' = V' \cup V_{test}$
    break
  Target nodes adjacent to outer nodes:
  Set $V_{inner} = V_{test} \backslash V_{outer}$
  Set $V_{added\_targets} = \{i \in V_{inner} | i \in \mathcal{N}_j' \text{ for some } j \in V_{outer}\}$
  Set $V' = V' \cup V_{added\_targets}$
  Update parameters for next loop:
  Set $V_{coverage} = \{i \in V_{test} | i \in \mathcal{N}_j' \text{ for some } j \in V_{added\_targets}\}$
  Set $V_{test} = V_{test} \backslash V_{coverage}$

distribute resources to these nodes uniformly:

$$B_i = \begin{cases} \frac{1}{|V'|}\mathcal{B}_b & \text{if } i \in V' \\ 0 & \text{else.} \end{cases}$$

Or we can distribute resources to more central nodes among those targeted:

$$B_i = \begin{cases} \frac{|\mathcal{N}_i| C_i}{\sum_{j \in V'} |\mathcal{N}_j| C_j}\mathcal{B}_b & \text{if } i \in V' \\ 0 & \text{else.} \end{cases}$$

We note that all of these heuristics are entirely based on network structure, and by design will symmetrically allocate resources in accordance with Proposition 3.1.11. Furthermore by limiting the number of nodes that are targeted, we limit the computational complexity of determining resource allocation. Though these network reductions may be impractical or provide minimal simplifications for certain network

compositions, these heuristics are generally quite flexible and can be employed in conjunction with other heuristics quite easily, as with the examples provided here. They can also be applied similarly for the curing models examined in Chapter 4.

3.4.3 Dense Networks

We now consider a limitation of the heuristics presented thus far. Our previous sets of solutions depend on the assumption that a network will possess a set of nodes whose respective neighbourhoods are strict subsets of some other node's neighbourhood (i.e., they assume $\exists i, j \in V$ that $\mathcal{N}_i' \subset \mathcal{N}_j'$). While this may be a useful tactic when considering sparser networks, like the one depicted in Figure 3.1, it is a very limiting requirement when we consider high density networks. There are even simple networks which fail to meet this requirement (e.g., consider the network in Figure 3.2). To account for this, we introduce a final node targeting strategy that does not have this limitation, but works based on similar principles.

Our previous methods worked by intentionally ignoring a subset of outer nodes, $V' = \{i \in V | \mathcal{N}_i' \subset \mathcal{N}_j', \text{for some } j \in V\}$. This was done with the idea of leveraging Proposition 3.1.7, in order to prevent suboptimal node targeting while still ensuring $\bar{B}_i > 0$, for every node $i \in V$. In cases where the conditions of Proposition 3.1.7 are not met, this technique is ineffective. Instead, we introduce an alternative method for defining inner and outer nodes.

This technique, presented in Algorithm 3, begins by ranking the nodes by some centrality measure (closeness centrality for our purposes), from highest to lowest. Nodes are added one-by-one to the set of targeted nodes, V', in order of rank. Once the set of targeted nodes provides total coverage of the network (i.e., the neighbourhoods

Algorithm 3 Node Targeting Algorithm for Dense Networks

$V \leftarrow$ set of all nodes
$V' \leftarrow$ nodes to be targeted
$C_i \leftarrow$ centrality score of given node

Without loss of generality assume $C_1 > C_2 > \cdots > C_N$, where N is the network size

Start with $V' = \emptyset$, $V_{coverage} = \emptyset$, n=0
while $V_{coverage} \neq V$ **do**
    Add most central node to target set:
    $V' = V' \cup \{n\}$
    Update loop parameters:
    Set $V_{coverage} = \{i \in V | i \in \mathcal{N}'_j, \text{ for some } j \in V'\}$
    n++

Optional:
for $n = |V'| : 1$ **do**
    Let $V_{coverage} = \{i \in V | i \in \mathcal{N}'_j, \text{ for some } j \in V' \backslash \{n\}\}$
    if $V_{coverage} = V$ **then**
        Set $V' = V' \backslash \{n\}$

of these nodes contains the entire network), the algorithm stops. As an additional reduction we can also go through the list of targeted nodes in reverse order and remove any nodes not required to maintain full coverage of the network. Similar to the previous heuristics, the objective is to target every node in the network (either directly or indirectly), while assigning resources to as few nodes as possible.

Like with our previous heuristics, once the target set has been determined, we can either distribute resources uniformly among these nodes:

$$B_i = \begin{cases} \frac{1}{|V'|}\mathcal{B}_b & \text{if } i \in V' \\ 0 & \text{else.} \end{cases}$$

Or we can distribute resources preferentially, according to centrality:

$$B_i = \begin{cases} \frac{|\mathcal{N}_i|C_i}{\sum_{j \in V'} |\mathcal{N}_j|C_j} \mathcal{B}_b & \text{if } i \in V' \\ 0 & \text{else.} \end{cases}$$

Chapter 4

Curing-Infection Problems

Another technique to control network infection for the Polya network model is to control the allocation of our curing parameters, $\{\Delta_{b,i}(n)\}_{i=1}^{N}$, subject to a fixed budget, $\mathcal{B}$. As in the initialization setup, this can be done for a finite or infinite horizon. Previously, [19] examined the problem of minimizing the limiting average infection rate and introduced a number of heuristic optimization policies. Some of these strategies were discussed briefly in Chapter 3. We begin by refocusing the problem in a game theory setup.

4.1 Game-Theoretic Setup

We consider a two player game, where player one minimizes the average infection rate, $\tilde{I}_n$, while player two maximizes this same value. Player one controls the distribution of curing parameters, $\{\Delta_{b,i}(n)\}_{i=1}^{N}$, in accordance with a budget, $\mathcal{B}_b$, while player two controls the distribution of the infection parameters, $\{\Delta_{r,i}(n)\}_{i=1}^{N}$ in accordance with a budget, $\mathcal{B}_r$. The budget in each case is fixed for all time steps, and the allocation of resources for a given time n is determined prior to any draws being made at that time. Thus, if either player allocates resources to a node for which the opposite colour

ball is drawn, those resources will go to waste.

Formally, player one's objective is to find:

$$\min_{\{\{\Delta_{b,i}(k)\}_{i=1}^{N}:\sum_{i=1}^{N}\Delta_{b,i}(k)=\mathcal{B}_b\},k=1,\ldots,n} \tilde{I}_n, \tag{4.1}$$

assuming the minimum exists, while player two's objective is to find:

$$\max_{\{\{\Delta_{r,i}(k)\}_{i=1}^{N}:\sum_{i=1}^{N}\Delta_{r,i}(k)=\mathcal{B}_r\},k=1,\ldots,n} \tilde{I}_n, \tag{4.2}$$

assuming the maximum exists. Together these constitute what we refer to as the Delta-curing problem.

For a general network, finding an optimal control policy for either player can be complicated, and the behaviour of $\tilde{I}_n$ is difficult to study. As discussed in [19], there is a strong correlation between the behaviour of $\tilde{S}_n$ and $\tilde{I}_n$, and thus we can simplify our problem by first considering a game on the expected network exposure for a given time step, rather than the limiting average infection rate. Using this measure as a proxy will drastically simplify the complexity of this analysis.

We consider the expected network exposure $E[\tilde{S}_n|\mathcal{F}_{n-1}] = E[\tilde{S}_n|\sigma(\{Z_j^{n-1}\}_{j=1}^{N})]$ as a two player zero-sum game, where player one minimizes the expected network exposure over the curing parameters $\{\Delta_{b,i}(n)\}_{i=1}^{N}$ and player two maximizes the same value over the infection parameters $\{\Delta_{r,i}(n)\}_{i=1}^{N}$. We begin with an important result about the expected network exposure.

Proposition 4.1.1. (Convexity-Concavity of Network Exposure): *Let $\mathcal{G} = (V, \mathcal{E})$ be a general network, and consider the Polya network contagion model on $\mathcal{G}$, with arbitrary initial conditions. Then, the expected network exposure $E[\tilde{S}_n|\mathcal{F}_{n-1}]$ is convex with respect to the curing parameters $\{\Delta_{b,i}(n)\}_{i=1}^{N}$ and concave with respect to the infection parameters $\{\Delta_{r,i}(n)\}_{i=1}^{N}$, for all n.*

Proof. Using (2.2) and (2.4), we consider $E[\tilde{S}_n|\mathcal{F}_{n-1}]$ as a function of the vectors

$$\mathbf{x} = (x_1, \ldots, x_N)^T = (\Delta_{b,1}(n), \ldots, \Delta_{b,N}(n))^T$$

$$\mathbf{y} = (y_1, \ldots, y_N)^T = (\Delta_{r,1}(n), \ldots, \Delta_{r,N}(n))^T,$$

by reformulating (2.4) as follows:

$$S_{i,n} = f_{i,n}(\mathbf{x}, \mathbf{y}, Z_n) = \frac{c_i + \delta_i(\mathbf{y}, Z_n)}{c_i + d_i + \sigma_i(\mathbf{x}, Z_n) + \delta_i(\mathbf{y}, Z_n)},$$

where

$$c_i = \bar{R}_i + \sum_{t=1}^{n-1} \sum_{j \in \mathcal{N}_i'} \Delta_{r,j}(t) Z_{j,t}$$

$$d_i = \bar{B}_i + \sum_{t=1}^{n-1} \sum_{j \in \mathcal{N}_i'} \Delta_{b,j}(t)(1 - Z_{j,t})$$

$$\delta_i(\mathbf{y}, Z_n) = \sum_{j \in \mathcal{N}_i'} y_j Z_{j,n}$$

$$\sigma_i(\mathbf{x}, Z_n) = \sum_{j \in \mathcal{N}_i'} x_j (1 - Z_{j,n}).$$

Alternatively, let A represent the adjacency matrix (including self connections) for our given network $\mathcal{G}$, where $A_{ij} = 1$ iff $i \in \mathcal{N}_j'$, and $A_{ij} = 0$, otherwise. Then construct an $N \times N$ square matrix, D, where $D_{ij} = A_{ij}(1 - Z_{j,n})$. Letting D_i represent the i^{th} row of the matrix D, we get $\sigma_i(x, Z_n) = D_i \mathbf{x}$, where

$$D_i \mathbf{x} = \begin{bmatrix} A_{i1}(1 - Z_{1,n}) & \cdots & A_{iN}(1 - Z_{N,n}) \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix}.$$

Likewise, we can construct an $N \times N$ square matrix, C, where $C_{ij} = A_{ij} Z_{j,n}$.

Letting C_i represent the i^{th} row of the matrix C, we get the following:

$$\delta_i(y, Z_n) = C_i\mathbf{y} = \begin{bmatrix} A_{i1}Z_{1,n} & \cdots & A_{iN}Z_{N,n} \end{bmatrix} \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}.$$

We thus can write

$$f_{i,n}(\mathbf{x}, \mathbf{y}, Z_n) = \frac{c_i + C_i\mathbf{y}}{c_i + d_i + C_i\mathbf{y} + D_i\mathbf{x}},$$

noting of course that the matrices C and D are functions of Z_n.

Taking the expectation of $S_{i,n}$ given the history of the contagion process up to time $n-1$, we get

$$\begin{aligned} E[S_{i,n}|\mathcal{F}_{n-1}] &= E[f_{i,n}(\mathbf{x}, \mathbf{y}, Z_n)|\mathcal{F}_{n-1}] \\ &= \sum_{z_n \in \{0,1\}^N} f_{i,n}(\mathbf{x}, \mathbf{y}, z_n) P(Z_n = z_n | Z^{n-1} = z^{n-1}). \end{aligned} \tag{4.3}$$

We note that $P(Z_n = z_n|Z^{n-1} = z^{n-1})$ is independent of our choice of $\mathbf{x}$, and for any fixed realization $z_n = (z_{1,n}, \ldots, z_{N,n}) \in \{0,1\}^N$, $f_{i,n}(\mathbf{x}, \mathbf{y}, z_n)$ is convex in x over $\mathbb{R}^N_{\geq 0}$ (the proof is given in [19]). Hence, it follows that $E[S_{i,n}|\mathcal{F}_{n-1}]$ is convex in $\mathbf{x}$ over $\mathbb{R}^N_{\geq 0}$, and thus so too is $E[\tilde{S}_n|\mathcal{F}_{n-1}]$. Concavity in $\mathbf{y}$ follows from a symmetry argument wherein we note

$$1 - f_{i,n}(\mathbf{x}, \mathbf{y}, Z_n) = \frac{d_i + D_i\mathbf{x}}{c_i + d_i + C_i\mathbf{y} + D_i\mathbf{x}}$$

is convex in $\mathbf{y}$, thus showing $f_{i,n}(\mathbf{x}, \mathbf{y}, Z_n)$ is concave in $\mathbf{y}$. The rest of the proof follows as in the convexity argument for $\mathbf{x}$. □

The natural symmetry of this model makes this result rather intuitive, and allows us to establish a nice setup for the game theoretic problem. We have proven that $E[\tilde{S}_n|\mathcal{F}_{n-1}]$ is convex in $\mathbf{x}$ and concave in $\mathbf{y}$ over $\mathbb{R}^N_{\geq 0} \times \mathbb{R}^N_{\geq 0}$. In order to get a better

understanding of this function, we consider the partial derivatives with respect to $\mathbf{x}$ and $\mathbf{y}$. We note that

$$\nabla_{\mathbf{x}} f_{i,n}(\mathbf{x}, \mathbf{y}, z_n) = \frac{-\nabla_{\mathbf{x}} D_i \mathbf{x}(c_i + C_i \mathbf{y})}{(c_i + d_i + C_i \mathbf{y} + D_i \mathbf{x})^2}$$
$$\nabla_{\mathbf{y}} f_{i,n}(\mathbf{x}, \mathbf{y}, z_n) = \frac{\nabla_{\mathbf{y}} C_i \mathbf{y}(d_i + D_i \mathbf{x})}{(c_i + d_i + C_i \mathbf{y} + D_i \mathbf{x})^2},$$

and furthermore

$$\frac{\partial}{\partial x_j} D_i \mathbf{x} = A_{ij}(1 - z_{j,n})$$
$$\frac{\partial}{\partial y_j} C_i \mathbf{y} = A_{ij} z_{j,n}.$$

Since A_{ij} and $z_{j,n}$ can only take values in $\{0, 1\}$, we get that $\frac{\partial}{\partial x_j} f_{i,n}(\mathbf{x}, \mathbf{y}, z_n) \leq 0$ and $\frac{\partial}{\partial y_j} f_{i,n}(\mathbf{x}, \mathbf{y}, z_n) \geq 0$. Using this fact in conjunction with Equation (4.3) we determine that over the space $\mathcal{X} \times \mathcal{Y} = \mathbb{R}^N_{\geq 0} \times \mathbb{R}^N_{\geq 0}$, $E[\tilde{S}_n | \mathcal{F}_{n-1}]$ has no saddle point. This lack of saddle point is demonstrated in Figure 4.1, which depicts the shape of $E[\tilde{S}_n | \mathcal{F}_{n-1}]$ for a single node network. However, given our fixed allocation budgets $\mathcal{B}_b$ and $\mathcal{B}_r$, we restrict ourselves to considering sets of the form $\mathcal{X} \times \mathcal{Y} = \{\{\Delta_{b,i}(n)\}_{i=1}^N \in \mathbb{R}^N_{\geq 0} | \sum_{i=1}^N \Delta_{b,i}(n) \leq \mathcal{B}_b\} \times \{\{\Delta_{r,i}(n)\}_{i=1}^N \in \mathbb{R}^N_{\geq 0} | \sum_{i=1}^N \Delta_{r,i}(n) \leq \mathcal{B}_r\}$.

Returning to our game over the expected network exposure, we remark that for any given n, the sets $\mathcal{X} = \{\{\Delta_{b,i}(n)\}_{i=1}^N \in \mathbb{R}^N_{\geq 0} | \sum_{i=1}^N \Delta_{b,i}(n) \leq \mathcal{B}_b\}$ and $\mathcal{Y} = \{\{\Delta_{r,i}(n)\}_{i=1}^N \in \mathbb{R}^N_{\geq 0} | \sum_{i=1}^N \Delta_{r,i}(n) \leq \mathcal{B}_r\}$ are convex and compact. This gives rise to the following result.

Theorem 4.1.2. (Nash Equilibrium for Network Exposure): *Let $\mathcal{G} = (V, \mathcal{E})$ be a general network equipped with the Polya network contagion model, with arbitrary initial conditions. For a given time n, consider a two-player zero-sum game where player one tries to minimize $E[\tilde{S}_n | \mathcal{F}_{n-1}]$ over the parameters $\{\Delta_{b,i}(n)\}_{i=1}^N$ and player two tries to maximize $E[\tilde{S}_n | \mathcal{F}_{n-1}]$ over the parameters $\{\Delta_{r,i}(n)\}_{i=1}^N$. Then, if*

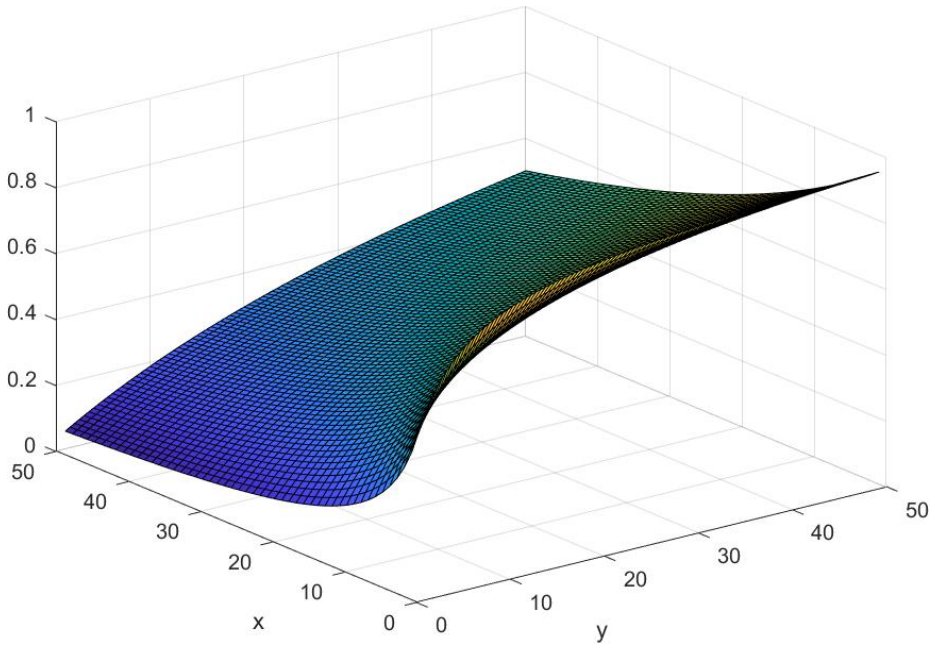


Figure 4.1: Sample plot of $E[\tilde{S}_n|\mathcal{F}_{n-1}]$ for a single node network containing equal parts red and black balls, over varying values of $x = \Delta_b$ and $y = \Delta_r$. A saddle point will not exist over $\mathbb{R}^N_{\geq 0} \times \mathbb{R}^N_{\geq 0}$ regardless of the network structure.

we take our set of allowable policies to be of the form $\mathcal{X} \times \mathcal{Y} = \{\{\Delta_{b,i}(n)\}_{i=1}^N \in \mathbb{R}^N_{\geq 0} | \sum_{i=1}^N \Delta_{b,i}(n) \leq \mathcal{B}_b\} \times \{\{\Delta_{r,i}(n)\}_{i=1}^N \in \mathbb{R}^N_{\geq 0} | \sum_{i=1}^N \Delta_{r,i}(n) \leq \mathcal{B}_r\}$*, the resulting game admits a Nash equilibrium . Moreover, the equilibrium policy will satisfy* $\sum_{i=1}^N \Delta_{b,i}(n) = \mathcal{B}_b$ *and* $\sum_{i=1}^N \Delta_{r,i}(n) = \mathcal{B}_r$*.*

Proof. Since the function is convex-concave and over a compact set, the existence follows from the classical minimax theorem, see [6]. By the definition of $E[\tilde{S}_n|\mathcal{F}_{n-1}]$, and since the function has no saddle point in the interior of its domain, the optimal policy will utilize the full budget. □

The equilibrium policy from Theorem 4.1.2 can be determined numerically using gradient descent algorithms [5] (as we are optimizing a convex/concave function over a simplex), but for large networks such algorithms can be computationally expensive,

especially when considering the complexity of $E[\tilde{S}_n|\mathcal{F}_{n-1}]$ as seen in (4.3).

From here we direct our focus on two main points. First, extending discussions for the expected network exposure, $E[\tilde{S}_n|\mathcal{F}_{n-1}]$, to the objective functions in (4.1) and (4.2). Additionally, we want to carry over our discussion from Chapter 3 to develop policies for the two player games on both the average infection rate, and the proxy measure of the expected network exposure.

4.2 Heuristic Strategies

When developing heuristics in the initialization case, we made heavy use of Proposition 3.1.7, which proved that any optimal solution to Problem 3.1.1 would allocate no resources to outer nodes (i.e., nodes with a nested neighbourhood). For the general Delta-curing problem, we have no such result, as it is possible to manufacture situations where the equilibrium policy as in Theorem 4.1.2 will assign resources to outer nodes. Such cases however arise under specific circumstances, and it is uncommon for these nodes to receive significant resources. Thus, we introduce a similar set of resource allocation policies for the game-theoretic setup as we did for the initialization problems in Chapter 3.

Note that while any of these resource allocation policies can be similarly applied to a player controlling the distribution of infection resources, for the purposes of our analysis here, we will consider these strategies from the perspective of the player controlling the curing resources.

4.2.1 Interior Node Targeting

As in Section 3.4.1 we want to first try targeting inner nodes within a given network. While Proposition 3.1.7 is not directly applicable to the setup as described in this chapter, the related allocation policies may still prove effective. We take the set of inner nodes to be $V' = \{i \in V | \mathcal{N}_i' \not\subset \mathcal{N}_j', \forall j \in V\}$ and we let $\mathcal{B}_b$ be the budget for curing resources $\{\Delta_{b,i}(n)\}_{i=1}^N$ at a given time n. As in the initialization case, our first allocation policy assigns resources according to the following scheme:

$$\Delta_{b,i}(n) = \begin{cases} \frac{1}{|V'|}\mathcal{B}_b & \text{if } i \in V' \\ 0 & \text{else.} \end{cases}$$

For this strategy, this policy is fixed for every time n. In Section 3.4.1 we introduced another strategy that accounted for the centrality of a given node. In this game-theoretic setup we also want to account for the evolution of the super urn proportion, $S_{i,n-1}$, of each node. As before we let C_i represent the closeness centrality of a given node i, calculated by (3.11). We then allocate resources as follows:

$$\Delta_{b,i}(n) = \begin{cases} \frac{|\mathcal{N}_i| C_i S_{i,n-1}}{\sum_{j \in V'} |\mathcal{N}_j| C_j S_{j,n-1}}\mathcal{B}_b & \text{if } i \in V' \\ 0 & \text{else.} \end{cases}$$

As in Chapter 3 we may choose to use replace closeness centrality with some alternate centrality measure.

4.2.2 Minimized Node Targeting

We now move on to strategies presented in Section 3.4.2. Again we want to implement the strategy of minimizing the number of targeted nodes. We also want to introduce a modification that accounts for the super urn proportion of each node. We let V'

represent the set of targeted nodes, as determined using Algorithm 2. We then use the following two resource allocation schemes:

$$\Delta_{b,i}(n) = \begin{cases} \frac{1}{|V'|}\mathcal{B}_b & \text{if } i \in V' \\ 0 & \text{else,} \end{cases}$$

which allocates the resources uniformly among the set of target nodes, or

$$\Delta_{b,i}(n) = \begin{cases} \frac{|\mathcal{N}_i|C_iS_{i,n-1}}{\sum_{j\in V'}|\mathcal{N}_j|C_jS_{j,n-1}}\mathcal{B}_b & \text{if } i \in V' \\ 0 & \text{else,} \end{cases}$$

which accounts for the centrality and super urn proportion of the targeted nodes.

4.2.3 Dense Networks

As discussed in Section 3.4.3, the previous strategies were limited in that the were only effective for sparse network setups. We thus employ the same techniques used prior for dense network setups. We let V' represent the set of target nodes as determined by Algorithm 3. Once again, from this target set, we allocate resources uniformly among these nodes:

$$\Delta_{b,i}(n) = \begin{cases} \frac{1}{|V'|}\mathcal{B}_b & \text{if } i \in V' \\ 0 & \text{else.} \end{cases}$$

Alternatively, we can distribute resources based on centrality and super urn proportion of the targeted nodes:

$$\Delta_{b,i}(n) = \begin{cases} \frac{|\mathcal{N}_i|C_iS_{i,n-1}}{\sum_{j\in V'}|\mathcal{N}_j|C_jS_{j,n-1}}\mathcal{B}_b & \text{if } i \in V' \\ 0 & \text{else.} \end{cases}$$

The effectiveness of the strategies described here, as well as the respective counterparts from Chapter 3, is analyzed further in Chapter 5.

Chapter 5

Simulation Results

We evaluate the performance of the heuristic policies detailed in Chapters 3 and 4, and compare their performance to supposed 'optimal' strategies obtained through gradient descent (note that we say 'optimal' because the gradient descent only converges to an optimal policy for the one-step proxy measures we introduced in both the initialization and curing cases). Algorithm 4 provides the general format of the simulations, while specifics of both the initialization and curing cases are provided in Sections 5.2 and 5.3, respectively.

In order to ensure effectiveness of different strategies for different network compositions, we test these policies for a few different networks, each of which is depicted in Figure 5.1. These networks include algorithmically generated 100 node Barabasi-Albert networks, depicted in Figures 5.1a and 5.1b, with average densities of 1.98 and 18.9 connections per node, respectively. The third network, depicted in Figure 5.1c, was generated by a tool [30] that crawled through posts in a Facebook group in order to establish connections between users based on interactions on user posts (either via commenting, or 'likes' on the post or comments). The resulting 1,363 node graph contains an average density of 3.56 connections per node, and provides a topology for

Algorithm 4 Simulation Setup

$A \leftarrow$ adjacency matrix of the network
numTrials $\leftarrow$ trials to run for given policy
steps $\leftarrow$ number of time steps for each trial
initPolicy $\leftarrow$ initialization policy (uniform in no control case)
curePolicy $\leftarrow$ Delta-curing policy (uniform in no control case)

Assign $(\mathbf{B}, \mathbf{R})$ using *initPolicy* (under initialization budget)
for s = 1 : *numTrials* **do**
    $\vec{Z}_s \leftarrow$ RunTrial$(A, \mathbf{R}, \mathbf{B}, \mathcal{B}_r, \mathcal{B}_b, steps, curePolicy)$

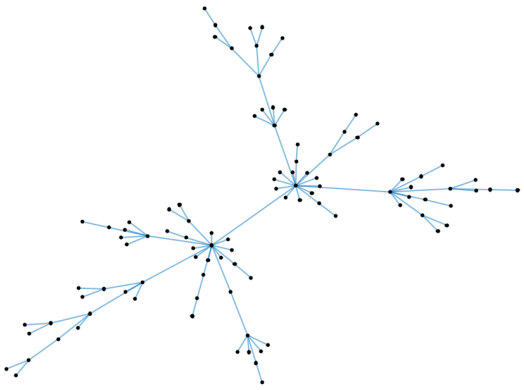
procedure RunTrial$(A, \mathbf{R}, \mathbf{B}, \mathcal{B}_r, \mathcal{B}_b, steps, curePolicy)$
    Initialize $S_{i,0}$ using R_i and B_i for all $i \in V$
    for t = 1 : *steps* **do**
        Assign $\Delta_{b,i}(t)$, $\Delta_{r,i}(t)$ using *curePolicy* (under curing budget)
        Generate $\vec{Y} \sim \mathbf{Uniform}([0,1])^N$
        if $Y_i \leq S_{i,t-1}$ **then**
            $Z_{i,t} = 1$
        else
            $Z_{i,t} = 0$
        Update $S_{i,t}$ using $\Delta_{r,i}$ and $\Delta_{b,i}(t)$ for all $i \in V$

a real-world social network.

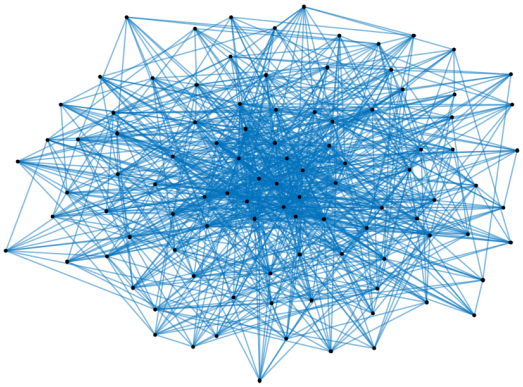
5.1 Targeting Algorithms

Since the majority of the following analysis applies curing policies based on targeting subsets of nodes, we want to provide a quick comparison of the different node targeting algorithms.

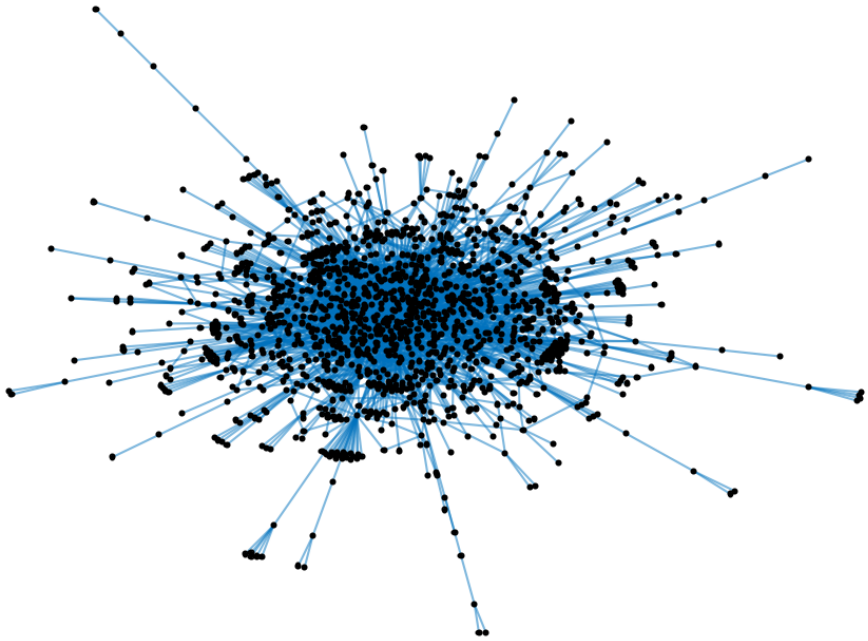
Figure 5.2 depicts a comparison of the different targeting algorithms for the sparse network seen in Figure 5.1a. For this specific network, we note that the target set in Figure 5.2b, is a subset of the set of inner nodes in Figure 5.2a, which is itself a subset of the target set from Figure 5.2c. While, this is not a general rule, it is a common



(a) Barabasi-Albert network with 100 nodes and 99 edges.



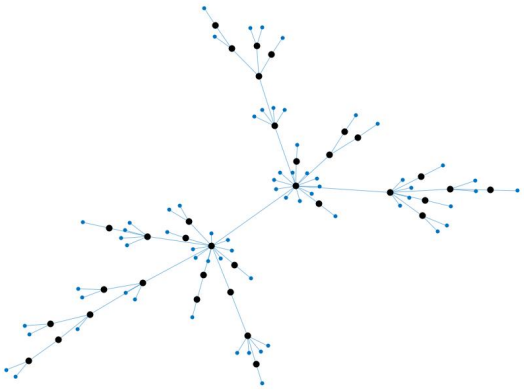
(b) Barabasi-Albert network with 100 nodes and 945 edges.



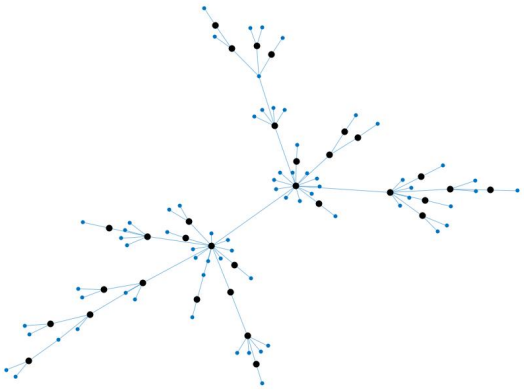
(c) Facebook network with 1,363 nodes and 2,425 edges.

Figure 5.1: Mid to large scale networks used for simulation purposes. Adjacency matrices for each of these networks is available online at https://bit.ly/2ygLEqg.

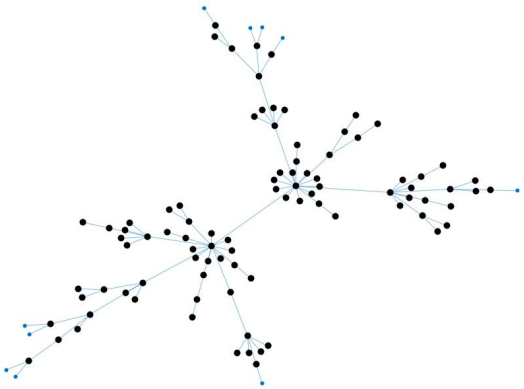
trend for very low-density networks. Algorithm 3 is not designed for use over sparse networks, and does not successfully leverage Proposition 3.1.7. Thus, we know that



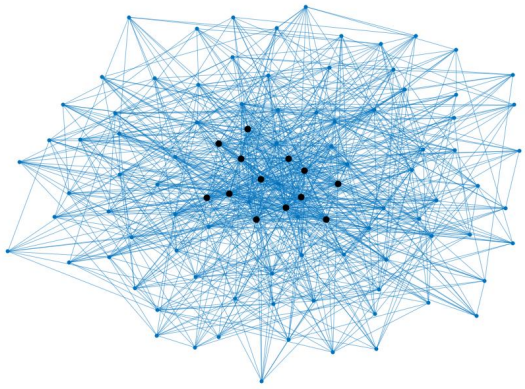
(a) Example set of inner nodes.



(b) Example target set using Algorithm 2.



(c) Example target set using Algorithm 3 for sparse networks.



(d) Example target set using Algorithm 3 for dense networks.

Figure 5.2: Comparison of target sets for various node targeting algorithms. In each case, the set of targeted nodes is enlarged and coloured black. It is important to note that Figure 5.2c, as depicted here, does not apply the optional component of Algorithm 3. In fact, if we instead choose to apply that option, we end up getting the same target set as shown in Figure 5.2b.

distributing resources among this set of target nodes will result in inefficiencies.

If we compare this to how Algorithm 3 applies to denser networks, as shown in Figure 5.2d, we observe a significant reduction in size of the set of target nodes. Algorithm 2, on the other hand, will be entirely ineffective at determing a subset of

Table 5.1: Initialization Strategies

(i)	Constrained gradient descent on a simplex: Find B_i using Algorithm 1 on the function $\tilde{I}_1$
(ii)	Uniformly allocate the budget to all nodes in the network: $B_i = \frac{\mathcal{B}_b}{N}$
(iii)	Uniformly allocate the budget to inner nodes in the network: $B_i = \begin{cases} \frac{1}{\lvert V' \rvert}\mathcal{B}_b & \text{if } i \in V' \\ 0 & \text{else} \end{cases}$, where $V' = \{k \in V \lvert \mathcal{N}'_k \not\subset \mathcal{N}'_j, \forall j \in V\}$
(iv)	Ratio of degree and closeness centrality for inner nodes in the network: $B_i = \begin{cases} \frac{\lvert\mathcal{N}_i\rvert C_i}{\sum_{j\in V'} \lvert\mathcal{N}_j\rvert C_j}\mathcal{B}_b & \text{if } i \in V' \\ 0 & \text{else} \end{cases}$, where $V' = \{k \in V \lvert \mathcal{N}'_k \not\subset \mathcal{N}'_j, \forall j \in V\}$
(v)	Allocate as in (iii) using V' from Algorithm 2
(vi)	Allocate as in (iv), using V' from Algorithm 2
(vii)	Allocate as in (iii) using V' from Algorithm 3 (without optional component)
(viii)	Allocate as in (iv), using V' from Algorithm 3 (without optional component)
(ix)	Ratio of degree and closeness centrality: $B_i = \frac{\lvert\mathcal{N}_i\rvert C_i}{\sum_{j\in V} \lvert\mathcal{N}_j\rvert C_j}\mathcal{B}_b$

nodes to target for this same network, due to the lack of nested nodes.

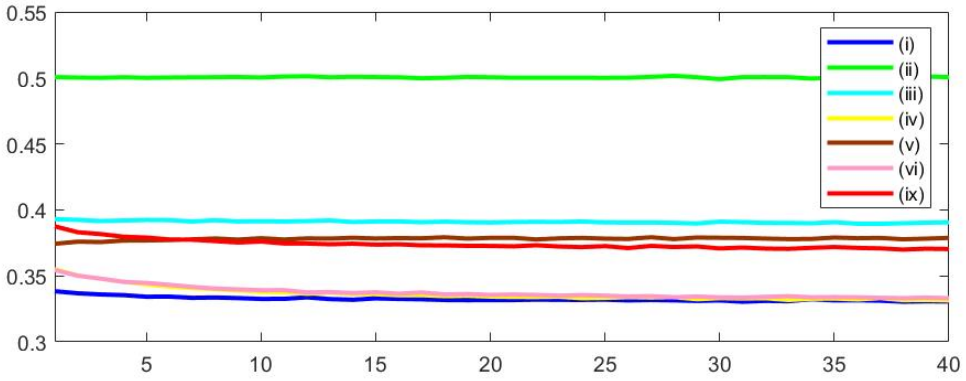
5.2 Initialization Trials

We start by comparing solutions to Problem 3.1.1, and evaluating their effectiveness for various network compositions. Table 5.1 provides an overview of the different initialization strategies we will be considering. More detailed explanations of these strategies can be found in Chapter 3. It is important to note that we do not include any trials that use the optional component of Algorithm 3, simply because we get the same target set as Algorithm 2 when applied to the networks in 5.1a and 5.1c, and it does not provide any further reduction to the target set when applied to the network in Figure 5.1b. The same reasoning applies to our curing strategies in Table 5.2.

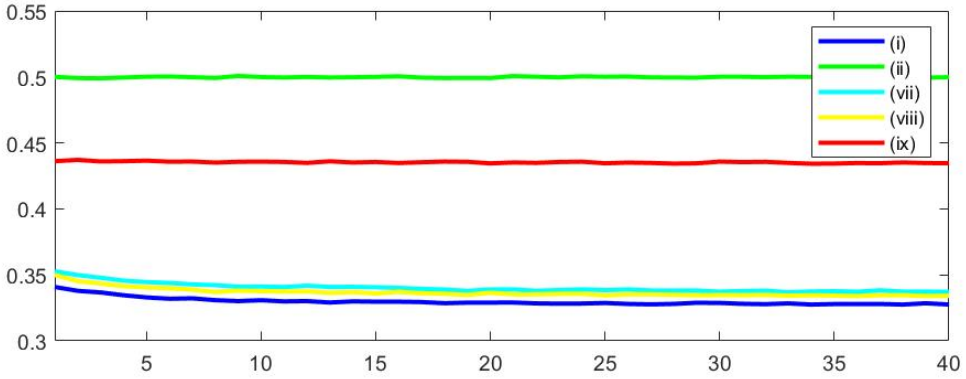
For each of the three networks in Figure 5.1 we follow the same process. We begin with equal initialization budgets for both red and black resources (i.e., take $\mathcal{B}_b = \mathcal{B}_r$), and let the infection initialization be uniform over the network (i.e., take $R_i = \frac{\mathcal{B}_r}{N}$, $\forall i \in V$). We choose our curing initialization $\mathbf{B}$ in accordance with the desired strategy from Table 5.1. Then, we run the Polya network contagion process with $\Delta_{r,i}(n) = \Delta_{b,i}(n) = \Delta$ fixed for each node $i \in V$ and each time n. We record the draw value, $Z_{i,n}$, for each node i at each time n. The draw values are averaged over the number of trials that are run and then averaged over the set of nodes in the network to obtain an empirical measure for the average infection rate $\tilde{I}_n$ at a given time n. The resulting output of these trials can be seen in Figure 5.3.

We start our comparison with a few obseravations. First, as expected, the solution determined using the gradient descent algorithm results in the lowest empirical average infection rate at time $n = 1$. For time $n > 1$, however, we see that this is not always the case. Figure 5.3c is of particular interest, as we clearly see that the gradient descent strategy begins with a much lower infection rate, but over time is surpassed by strategy (vi), which targets a reduced subset of nodes in accordance with Algorithm 2, with resources being focused more around central nodes.

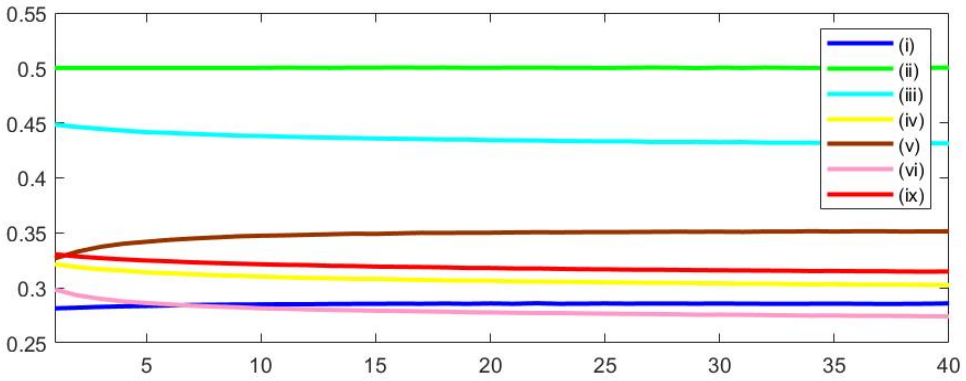
The fact that the best strategy at time $n = 1$ is not optimal as n increases supports our hypothesis that the optimal solution will change for increasing values of n and as $n \to \infty$. It is important to note, however, that our gradient algorithm merely converges to the optimal solution at time $n = 1$, and thus the solution we get from using this algorithm is not necessarily optimal, as the algorithm has not been run indefinitely. In fact, we can note that had we chosen to either use a finer filter when implementing the limit minimization rule, as described in Algorithm 1, or increased



(a) Results of initialization trials for low density Barabasi-Albert network depicted in 5.1a



(b) Results of initialization trials for high density Barabasi-Albert network depicted in 5.1b



(c) Results of initialization trials for Facebook network depicted in 5.1c

Figure 5.3: Plots of empirical average infection rate, $\tilde{I}_n$, for various initialization strategies across multiple networks. Lower values indicate lower levels of infection. In each case the network was given a fixed initialization budget of $\mathcal{B}_r = \mathcal{B}_b = 10N$, with $R_i = 10 \ \forall i \in V$, and the Polya network process was run over 1000 trials with a fixed $\Delta = 5$.

the stopping time, or even chosen a step size that allowed for faster convergence, we would have seen an improved performance using this strategy (at least at time $n = 1$).

Regardless, we consistently see that targeting a reduced subset of nodes, whether via Alogorithms 2 or 3, results in performance that is close to or better than that obtained using gradient descent. These strategies, (vi) and $(viii)$, always provide the best results of all the heuristic strategies being compared. This illustrates that we obtain near optimal performance while only distributing resources amongst a greatly reduced subset of nodes. Performance between the other heuristic strategies varies greatly depending on the network, though we note that even just targeting the inner nodes using strategy (iii) sees a significant improvement over the no control case (i.e., the results obtained using a uniform distribution of initialization resources).

We also remark that the general curve for each of these trials remains relatively flat. This is mostly due to the fact that Δ is assigned a fixed value. At a given time for a given node the expected number of red balls added to that particular node's super urn will be proportional to the average super urn proportion of the neighbouring nodes. This causes the super urn proportion of a given node to trend towards the average super urn proportion of its neighbours, which often takes a similar value because these nodes have overlapping sets of neighbours.

We next examine whether these same results carry over to the general curing case, or whether there are any notable differences. We should note that the curing and initialization policies are by nature not identical. They do, however, mirror one another in such a way that we can at least provide a surface-level comparison between the two setups.

Table 5.2: Curing Strategies

(i)	Constrained gradient descent on a simplex: Find $\Delta_{b,i}(n)$ using Algorithm 1 on the function $E[\tilde{S}_n \vert \mathcal{F}_{n-1}]$
(ii)	Uniformly allocate the budget to all nodes in the network: $\Delta_{b,i}(n) = \frac{\mathcal{B}_b}{N}$
(iii)	Uniformly allocate the budget to inner nodes in the network: $\Delta_{b,i}(n) = \begin{cases} \frac{1}{\lvert V' \rvert}\mathcal{B}_b & \text{if } i \in V' \\ 0 & \text{else} \end{cases}$, where $V' = \{k \in V \vert \mathcal{N}'_k \not\subset \mathcal{N}'_j, \forall j \in V\}$
(iv)	Ratio of degree, closeness centrality, and super urn proportion for inner nodes: $\Delta_{b,i}(n) = \begin{cases} \frac{\lvert\mathcal{N}_i\rvert C_i S_{i,n-1}}{\sum_{j\in V'} \lvert\mathcal{N}_j\rvert C_j S_{j,n-1}}\mathcal{B}_b & \text{if } i \in V' \\ 0 & \text{else} \end{cases}$, where $V' = \{k \in V \vert \mathcal{N}'_k \not\subset \mathcal{N}'_j, \forall j \in V\}$
(v)	Allocate as in (iii) using V' from Algorithm 2
(vi)	Allocate as in (iv), using V' from Algorithm 2
(vii)	Allocate as in (iii) using V' from Algorithm 3 (without optional component)
(viii)	Allocate as in (iv), using V' from Algorithm 3 (without optional component)
(ix)	Ratio of degree, closeness centrality and super urn proportion: $B_i = \frac{\lvert\mathcal{N}_i\rvert C_i S_{i,n-1}}{\sum_{j\in V} \lvert\mathcal{N}_j\rvert C_j S_{j,n-1}}\mathcal{B}_b$

5.3 Curing Trials

We consider the one-sided curing problem of trying to find a solution for (4.1), for some time n. As in the initialization case, we want to compare the effectiveness of the various curing policies presented in Chapter 4. A brief overview of these policies is found in Table 5.2.

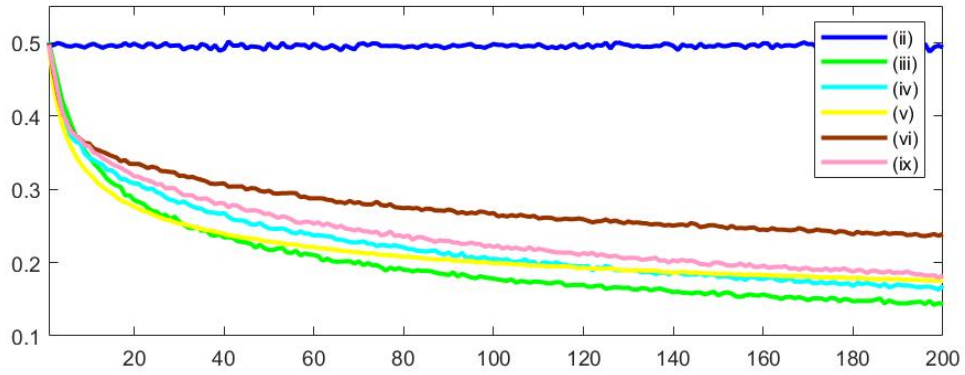
We begin with a uniform initialization, where each node is allocated a fixed number of red balls, R, and black balls, B, with $R = B$. We then run the Polya network contagion process. Before each draw is performed, we assign a number of curing resources to each node $\Delta_{b,i}(n)$ in accordance with the desired policy from Table 5.2, while the infection resources are assigned uniformly, $\Delta_{r,i}(n) = \frac{\mathcal{B}_r}{N}$ (with a fixed budget $\mathcal{B}_b = \mathcal{B}_r$ for each time n). As in the initialization case, we record the draw value, $Z_{i,n}$

for each node $i \in V$ at each time n. The draw values are averaged over the number of trials and the set of nodes to obtain a measure for the empirical average infection rate, $\tilde{I}_n$. The results of these simulations are depicted in Figure 5.4.

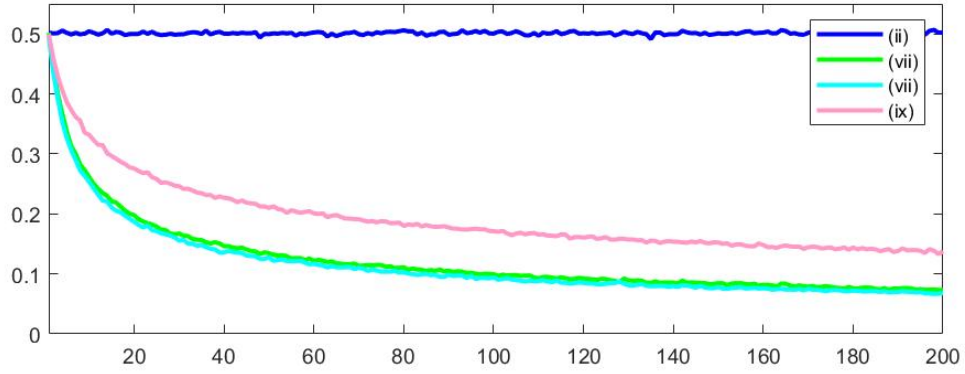
The first thing we remark is that unlike the initialization case, targeting a reduced subset of nodes based on centrality does not always lead to the best performance. This is evidenced by both Figures 5.4a and 5.4c, where strategy (vi) appears to perform worst of all the heuristics apart from the no control case. In fact, between the three network setups, the heuristics vary wildly in performance, with the best performing heuristic from [16] performing extremely well in Figure 5.4c, but performing significantly worse than the other techniques in 5.4b. The various strategies also perform to different degrees of success for different values of n (see strategy (vi) in Figure 5.4c).

Between the three networks, the strategies that seem to perform the most consistently well are strategies that target only the inner set of nodes (strategies (iii), (iv) and the analogous strategies for denser networks, (vii) and $(viii)$). It is interesting that these strategies would perform most consistently for the Delta-curing problem, but would perform far from optimally for the one-time initialization policy as seen in Figure 5.3. The main difference between these setups is that the solutions to the Delta-curing problem are applied continually as the infection evolve, while solutions to the initialization problem are applied once. Obviously, some of the benefits of targeting a greatly reduced set of nodes, as in seen in Figure 5.2b, are lost when they do not properly account for changes in the urn compositions. Interestingly, for denser networks, targeting a greatly reduced set of nodes as in Figure 5.2d was still very effective, most likely as a direct result of the high network density.

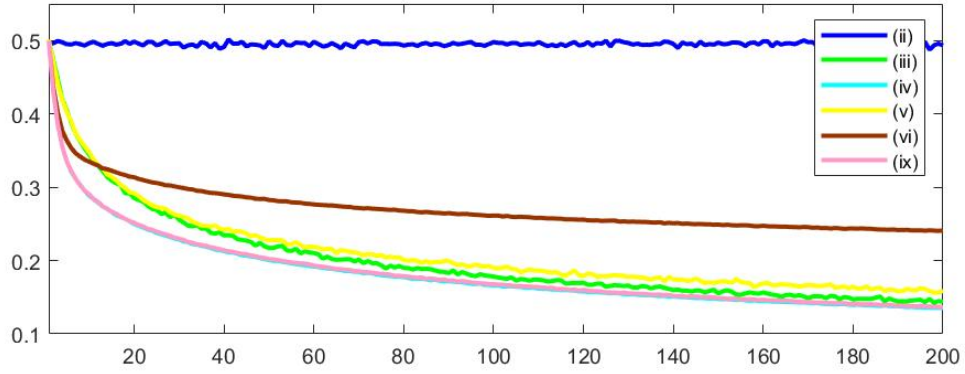
One notable exclusion from Figure 5.4 is the fact that none of the three subplots



(a) Results of curing trials for low density Barabasi-Albert network depicted in 5.1a



(b) Results of curing trials for high density Barabasi-Albert network depicted in 5.1b



(c) Results of curing trials for Facebook network depicted in 5.1c

Figure 5.4: Plots of empirical average infection rate, $\tilde{I}_n$, for various curing strategies across multiple networks. Lower values indicate lower levels of infection. In each case the network was initialized with $R_i = B_i = 10$, $\forall i \in V$, and the Polya network process was run over 500 trials with a fixed budget $\mathcal{B}_r = \mathcal{B}_b = 10N$, with $\Delta_r = 10$ fixed.

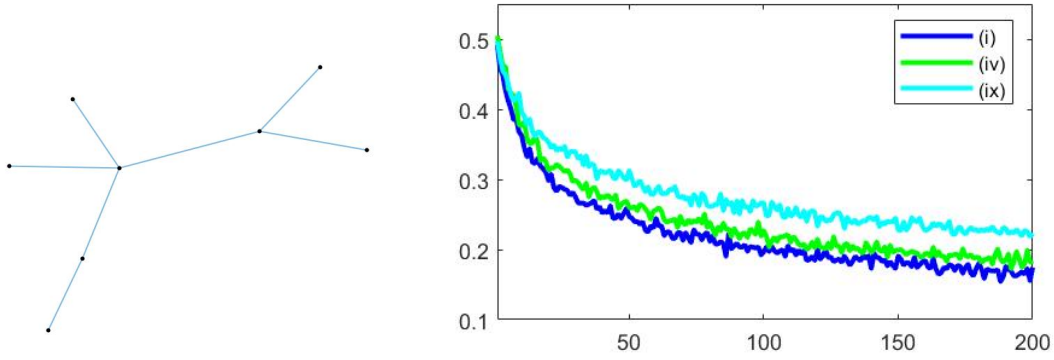


Figure 5.5: Empirical average infection rate, $\tilde{I}_n$, for small 8 node test network. The performance of gradient descent over the expected network exposure, $E[\tilde{S}_n|\mathcal{F}_{n-1}]$, was compared to the two best performing heuristics from Figure 5.4a. The network was initialized with $R_i = B_i = 10$, $\forall i \in V$, and the Polya network process was run over 500 trials with a fixed budget $\mathcal{B}_r = \mathcal{B}_b = 80$, with $\Delta_r = 10$ fixed.

show the performance obtained using gradient descent, in part due to the highly complex nature of the expected network exposure, $E[\tilde{S}_n|\mathcal{F}_{n-1}]$, which is the function over which we are minimizing. While any large scale gradient descent simulations are somewhat infeasible as a result, it is still possible to provide a comparison on a smaller scale, in order to see how far from 'optimal' our various heuristics are (note we say 'optimal' since the gradient descent algorithm only converges to an optimal solution for the one-step problem of minimizing $E[\tilde{S}_n|\mathcal{F}_{n-1}]$ over the curing resources, as we have discussed prior). We provide such a comparison in Figure 5.5. While it is clear that the gradient descent does marginally improve the empirical average infection rate, we see that strategy (iv) still performs comparably, at a fraction of the computational cost.

Chapter 6

Conclusion

In this work we considered the Polya contagion model and formulated control policies for this model designed to manipulate the spread of infection. We framed these problems in two lights: both as a one-time initialization problem, and a continual control problem that was an extension of the contagion curing problem presented in [19]. We provided simplified metrics for the one-sided finite-horizon version of these problems and proved the existence of optimal policies. We also provided heuristics that functioned primarily by modifying the scope of solutions to only a limited number of nodes. We then compared these heuristics to the one-step optimal policies that were obtained through use of gradient descent.

We established underlying results for the initialization problem proving that optimal initialization strategies would only allocate resources to inner nodes. We showed how network symmetry impacts the allocation of resources for optimal policies, and we presented empirical evidence to support the claim that nodes that are similar under some automorphism subgroup will be allocated the same number of resources. We also demonstrated that the heuristics that we developed for this problem will perform comparably to the one-step optimal policy in certain situations.

We framed the Delta-curing problem as a two-person zero sum game on the expected network exposure, and we proved the existence of a Nash equilibrium for this problem setup. We borrowed ideas from the initialization problems in order to develop heuristics for the Delta-curing problem and demonstrated that these strategies can perform near-optimally at a fraction of the computational expense.

There are numerous other frameworks with which the Polya network process can be studied, as well as further analysis that can be performed for the problems as presented in this work. This extends to the study of both infinite and finite horizon versions of these problems, as well as a deeper analysis of the two-sided initialization problem that was presented. The initialization problem can also be framed in a game-theoretic sense as was done for the Delta-curing problem, and the existence of a deterministic Nash equilibrium for this two-player game is not yet proven.

The Polya network process itself can also be modified for further analysis from a finite-memory perspective. It is possible to consider this process with an unknown network topology, or to consider the case where the underlying resource distribution is hidden from the observer. Further study of the network models in Chapter 2 may also prove useful in providing insight into the behaviour of the Polya network process itself, and re-formulating the initialization or Delta-curing problems in context of these models may be an interesting area of analysis. It may also be beneficial to develop a more in-depth comparison of this model to the common class of SIS models that exist in literature and to harden the applicability of this model for practical use.

Bibliography

[1] E. Adar and L. A. Adamic. Tracking information epidemics in blogspace. In *Proc. IEEE/WIC/ACM Int. Conf. Web Intelligence*, pages 207–214, 2005.

[2] F. Alajaji and T. Fuja. A communication channel modeled on contagion. *IEEE Trans. Inf. Theory*, 40(6):2035–2041, 1994.

[3] R. Albert and A. L. Barabási. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(1):47–97, 2002.

[4] R. Ash and C. Doléans-Dade. *Probability and Measure Theory*. 2000.

[5] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, MA, 2nd edition, 1999.

[6] D. P. Bertsekas, A. Nedić, and A. E. Ozdaglar. *Convex Analysis and Optimization*. Athena Scientific, Belmont, MA, 1st edition, 2003.

[7] N. Biggs. *Algebraic Graph Theory*. Cambridge University Press, 2 edition, 1994.

[8] M. Chen and M. Kuba. On generalized Polya urn models. *Journal of Appl. Probab.*, 50(4):1169–1186, 2013.

[9] F. Chung, S. Handjani, and D. Jungreis. Generalizations of Polya's urn problem. *Ann. Combin.*, 7:141–153, 2003.

[10] D. Easley and J. Kleinberg. *Networks, Crowds and Markets: Reasoning about a Highly Connected World*. Cambridge Univ. Press, 2010.

[11] F. Eggenberger and G. Polya. Über die Statistik verketteter Vorgänge. *Z. Angew. Math. Mech.*, 3(4):279–289, 1923.

[12] A. Fazeli and A. Jadbabaie. On consensus in a correlated model of network formation based on a Polya urn process. 2011.

[13] M. Garetto, W. Gong, and D. Towsley. Modeling malware spreading dynamics. In *Proc. Conf. Comp. Comm.*, volume 3, pages 1869–1879, 2003.

[14] C. D. Godsil and G. F. Royle. *Algebraic Graph Theory*, volume 207 of *Graduate Texts in Mathematics*. Springer, 2001.

[15] G. R. Grimmett and D. R. Stirzaker. *Probability and Random Processes*. Oxford Univ. Press, 3 edition, 2001.

[16] M. Hayhoe. A Polya urn stochastic model for the analysis and control of epidemics on networks. Master's thesis, Queen's University, 2017.

[17] M. Hayhoe, F. Alajaji, and B. Gharesifard. A Polya urn-based model for epidemics on networks. *Proc. 2017 American Cont. Conf.*, 2017.

[18] M. Hayhoe, F. Alajaji, and B. Gharesifard. A Polya contagion model for networks. *IEEE Trans. Cont. Netw. Sys.*, 5:1998–2010, 2018.

[19] M. Hayhoe, F. Alajaji, and B. Gharesifard. Curing epidemics on networks using a Polya contagion model. *IEEE/ACM Transactions on Networking*, 27(5):2085–2097, 2019.

[20] A. R. Hota and S. Sundaram. Game-theoretic protection against networked SIS epidemics by human decision-makers. *IFAC-PapersOnLine*, 51, 03 2017.

[21] L. Kim, M. Abramson, K. Drakopoulos, S. Kolitz, and A. Ozdaglar. Estimating social metwork structure and propagation dynamics for an infectious disease. In *Proc. Int. Conf. Social Computing, Behavioral-Cultural Modeling, and Prediction*, pages 85–93. Springer, 2014.

[22] S. Kudose. Equitable partitions and orbit partitions. 2009.

[23] P. Van Mieghem, J. Omic, and R. Kooij. Virus spread in networks. *IEEE/ACM Trans. Netw.*, 17(1):1–14, 2009.

[24] J. Nash. Equilibrium points in n-person games. In *Proceedings of the National Academy of Sciences USA*, volume 36, pages 48–49, 1950.

[25] J. Nash. Non-cooperative games. *The Annals of Mathematical Statistics*, 54:286–295, 1951.

[26] C. Nowzari, V. M. Preciado, and G. J. Pappas. Analysis and control of epidemics: A survey of spreading processes on complex networks. *IEEE Control Systems Magazine*, 36:26–46, 2016.

[27] P. E. Paré, J. Liu, C. L. Beck, A. Nedić, and T. Başar. Multi-competitive viruses over static and time-varying networks. In *2017 American Control Conference (ACC)*, pages 1685–1690, May 2017.

[28] G. Polya. Sur quelques points de la théorie des probabilités. *Annales de l'institut Henri Poincaré*, 1(2):117–161, 1930.

[29] G. Polya and F. Eggenberger. Sur l'interprétation de certaines courbes de fréquences. *Comptes Rendus C. R.*, 187:870–872, 1928.

[30] B. Rieder. Studying Facebook via data extraction: the Netvizz application. *Proc. 5th ACM Web Science Conf.*, pages 346–355, 2013.

[31] E. M. Rogers. *Diffusion of Innovations*. 5 edition, 2003.