# A Polya Urn Stochastic Model for the Analysis and Control of Epidemics on Networks

by

Mikhail Hayhoe

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
August 2017

# Abstract

This thesis introduces a model for epidemics on networks based on the classical Polya process. Temporal contagion processes are generated on the network nodes using a modified Polya sampling scheme that accounts for spatial infection among neighbouring nodes. The stochastic properties and asymptotic behaviour of the resulting network Polya contagion process are analyzed. Given the complicated nature of this process, three classical Polya processes, one computational and two analytical, are proposed to statistically approximate the contagion process of each node, demonstrating a good fit for a range of system parameters. An optimal control problem is formulated for minimizing the average infection using a limited curing budget, and a number of different curing strategies are presented, including a proven convergent gradient descent algorithm. The feasibility of the problem is proven under high curing budgets by deriving conservative lower bounds that turn some processes into supermartingales. Extensive simulations run on large-scale networks demonstrate the effectiveness of our proposed strategies.

# Acknowledgments

I would like to express my deep gratitude to my supervisors, Prof. Fady Alajaji and Prof. Bahman Gharesifard, for their guidance, mentoring, and support throughout my time at Queen's and especially during my Master's research. It was Prof. Alajaji's first-year linear algebra course that set me on the academic path I am on today, and it was under Prof. Gharesifard's supervision of my Honour's thesis that I discovered my interest in research.

I am sincerely grateful to my parents Mark and Carylin, my sister Linnea, and my grandparents Keith and Caryl for their unconditional love and support. I would also like to thank my friends, without whom I would not be the person I am today. Thanks to all of you.

Finally, I wish to acknowledge the Centre for Advanced Computing at Queen's University, whose computing cluster allowed for the execution of many of the simulation results presented herein.

# Contents

# List of Figures

# Chapter 1

# Introduction

In this work, we examine the dynamics and properties of a contagion process, or *epidemic*, on a network. Here an epidemic can represent a disease [31], a computer virus [22], the spread of an innovation, rumour or idea [44], or the dynamics of competing opinions in a social network [1].

Epidemics on networks have been intensively studied in recent years; see [33, 34, 30] and references therein and thereafter. Many different models for the study of infection propagation and curing exist in the literature. Our model is similar to the well-known susceptible-infected-susceptible (SIS) compartmental infection model [18], in the sense that initially, all nodes may be healthy or infected and as the epidemic spreads, nodes that are infected can be cured to become healthy, but any healthy node may become infected at any time, regardless of whether they have been cured previously. This compartmental relationship is illustrated in Figure 1.1. However, the dynamics of the traditional SIS model tend to be complicated, and often deterministic approximation methods are employed to simplify the analysis [34].

In contrast to the traditional SIS model, our model is motivated by the classical Polya contagion process, which evolves by sampling from an urn containing a finite

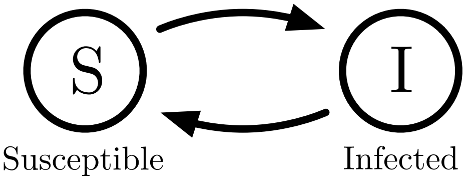


Figure 1.1: Susceptible-infected-susceptible compartmental model.

number of red and black balls [19, 39, 38]. The classical Polya model has been used and naturally arises in a wide range of applications, such as consensus dynamics [20] and population genetics [27]; see [37] for a summary. In the network Polya contagion model, each node of the underlying network is equipped with an individual urn; however, instead of sampling from these urns when generating its contagion process, each node has a "super urn", created by combining the contents of its own urn with those of its neighbours' urns. This adaptation captures the concept of spatial infection, since having infected neighbours increases the chance that an individual is infected in the future. This concept of the super urn sampling mechanism for incorporating spatial interactions was originally introduced in [11] in the context of the image segmentation and labeling problem. In this work, we investigate the dynamics and control of the resulting contagion process affecting each node of the network.

More specifically, we study the time evolution and stochastic properties of the proposed network contagion process. We derive an expression for the temporal $n$-fold joint probability distribution of the process. We show that this process, unlike the classical Polya urn process, is in general non-stationary, and hence not exchangeable. For the special case of complete networks, we analytically find the 1-dimensional and 2-dimensional $(n, 1)$-step marginal distributions of the contagion process. These results show that, even though it is not stationary, the process is nevertheless identically distributed with its latter two marginal distributions being invariant to time shifts. The process with finite memory is investigated, showing that the entire state is Markovian and the individual processes are quasi-Markovian in the entire state. We also establish several martingale properties regarding the network urn compositions, proving that the proportions of red balls in each node's urn as well as the network average urn proportion converge almost surely to a limit as time grows without bound. We next provide three approximations to the network contagion process by modelling each node's contagion process via a classical stationary Polya process [38]. In the first one, we approximate each node's process with the classical Polya process whose correlation parameter is empirically selected so that the Kullback-Leibler divergence measure between its $n$-fold joint distribution and that of the original node process is minimized. In the second approximation, we propose an analytical model whose parameters are chosen by matching its first and $(n, 1)$-step second-order statistics with those of the original node process, which fits well for large networks. The last approximation uses a classical Polya model with parameters chosen analytically that we show fits well for small networks. Simulation results are presented to support the validity of these approximations.

We further propose various natural ways to measure the total infection in the network Polya contagion model, and examine conditions under which these measures admit limits. Using these measures, we pose an optimal control problem within the context of the network Polya contagion model. We characterize bounds on the allocation of curing to individual nodes which turn the network Polya contagion process infection measures into supermartingales. Our result hence provides a conservative strategy for curing network epidemics. We next focus on realistic scenarios, where the curing budget is constrained. As our next contribution, we prove that the constrained gradient flow method is convergent for this problem and hence can be employed to find near-optimal strategies under a fixed curing budget at each time step. In spite of its effectiveness, as we demonstrate, the gradient flow strategy is computationally expensive and is only implementable in a centralized manner. These shortcomings motivate us to look into alternative strategies, which take advantage of notions of node centrality of the underlying network along with the composition of super urns at each time step. These strategies are less expensive computationally and can be adopted for implementation in a decentralized manner. Through extensive simulation results, we show that our proposed heuristic strategies perform well in curing epidemics.

The remainder of this thesis is organized as follows. Chapter 2 describes the network Polya contagion process, including its stochastic properties, some theoretical results giving conditions under which some processes become martingales, and approximations that can be used to estimate the limiting behaviour. Parts of this chapter appeared in [24] and [25]. Chapter 3 outlines the specific control problem to be considered in curing epidemics using the model, and discusses a number of theoretical, numerical and heuristic strategies to control the infection; minor parts of this chapter are present in [25]. Chapter 4 describes the simulations used to observe the performance of the model, and discusses the results in detail. Finally, Chapter 5 concludes the thesis.

## 1.1 Preliminaries

Let $(\Omega, \mathcal{F}, P)$ be a probability space, and consider the stochastic process $\{Z_n\}_{n=1}^{\infty}$, where each $Z_n$ is a random variable on $\Omega$. For a sequence $Z_i = (Z_{i,1}, ..., Z_{i,n})$, we use the notation $Z_{i,s}^{t}$ with $1 \leq s < t \leq n$ to denote the vector $(Z_{i,s}, Z_{i,s+1}, ..., Z_{i,t})$. We often refer to the indices of the process as "time" indices. Our technical results rely on notions from probability theory and stochastic processes, some of which we recall here. Precise definitions of all concepts, including that of *ergodicity*, can be found in standard texts (e.g., [8, 23]).

**Definition. (Filtration):** *The sequence of $\sigma$-algebras $\{\mathcal{F}_n\}_{n=1}^{\infty}$ is a filtration on the process $\{Z_n\}_{n=1}^{\infty}$ if*

- *$\mathcal{F}_n \subseteq \mathcal{F}$ for all $n$,*
- *$t \leq n \Rightarrow \mathcal{F}_t \subseteq \mathcal{F}_n$, and*
- *$Z_n$ is measurable with respect to $\mathcal{F}_n$ for all $n$.*

Herein we will consider a special kind of filtration on $\{Z_n\}_{n=1}^{\infty}$, called the natural filtration $\{\mathcal{F}_n^Z\}_{n=1}^{\infty}$, which informally is 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 on $\{Z_n\}_{n=1}^{\infty}$.

**Definition. (Stationary):** *The process $\{Z_n\}_{n=1}^{\infty}$ is stationary if its joint probability distribution is invariant to time shifts, i.e.,*

$$P(Z_1 = a_1, \ldots, Z_n = a_n) = P(Z_{1+s} = a_1, \ldots, Z_{n+s} = a_n)$$

*for all $n, s \in \mathbb{Z}_{\geq 1}$.*

**Definition. (Exchangeable):** *The process $\{Z_n\}_{n=1}^{\infty}$ is exchangeable if its joint probability distribution is invariant to permutations in the time index, i.e.,*

$$P(Z_{1+s} = a_1, \ldots, Z_{n+s} = a_n) = P(Z_{i_1} = a_1, \ldots, Z_{i_n} = a_n)$$

*for all $n, s \in \mathbb{Z}_{\geq 1}$ and every permutation $\{i_1, \ldots, i_n\}$ of $\{1+s, \ldots, n+s\}$.*

It directly follows from the definitions that an exchangeable process is stationary. We now describe a special kind of process, called a martingale.

**Definition. (Martingale):** *The process $\{Z_n\}_{n=1}^{\infty}$ is called a martingale (resp. submartingale, supermartingale) with respect to the process $\{Y_n\}_{n=1}^{\infty}$ if*

1. *$E[|Z_n|] < \infty$, and*
2. *for any $n \in \mathbb{Z}_{\geq 1}$, we have*
   *$E[Z_{n+1}|Y_n] = Z_n$,*
   *resp. $E[Z_{n+1}|Y_n] \geq Z_n$,*
   *resp. $E[Z_{n+1}|Y_n] \leq Z_n$, almost surely.*

If the inequalities in the above definition are made to be strict, we would call the resulting process a *strict submartingale* or *strict supermartingale*. We now recall the martingale convergence theorem, which states that any martingale (or submartingale, or supermartingale) converges almost surely to a random variable.

**Theorem 1.1.1. (Submartingale Convergence Theorem [8, Thm 6.4.3]):** *Let $\{Z_n\}_{n=1}^{\infty}$ be a submartingale. If*

$$\sup_{1 \leq n \leq \infty} E[Z_n] < \infty$$

*then there is some random variable $Z_\infty$ such that $Z_n \to Z_\infty$ almost surely.*

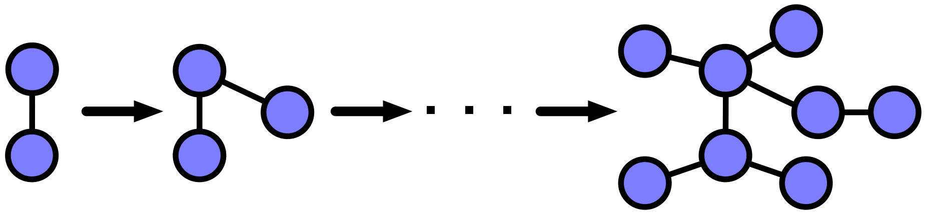


Figure 1.2: Illustration of the generation of a Barabasi-Albert network. We start with a 2-node complete network and add 6 nodes each with 1 edge.

**Corollary 1.1.2. (Supermartingale Convergence Theorem [8, Cor 6.4.4]):** *Let $\{Z_n\}_{n=1}^{\infty}$ be a supermartingale. If*

$$\inf_{1\leq n\leq\infty} E[Z_n] > -\infty$$

*then there is some random variable $Z_\infty$ such that $Z_n \to Z_\infty$ almost surely.*

Note that in particular a martingale is a submartingale, and hence Theorem 1.1.1 gives conditions for martingales to converge as well.

Two other types of closely-related processes we will consider are Markovian and quasi-Markovian processes.

**Definition. (Markov process with memory $M$):** *A process $\{Z_n\}_{n=1}^{\infty}$ is Markovian with memory $M$ if it satisfies the Markov property:*

$$P\left(Z_n = a_n \mid \{Z_t = a_t\}_{t=1}^{n-1}\right) = P\left(Z_n = a_n \mid \{Z_t = a_t\}_{t=n-M}^{n-1}\right) \quad \textit{for all } n > M.$$

*In other words, $\{Z_n\}_{n=1}^{\infty}$ depends (conditionally) only on the last $M$ steps.*

**Definition. (Quasi-Markovian):** *An individual process $\{Z_{i,n}\}_{n=1}^{\infty}$ from a process $\{Z_n\}_{n=1}^{\infty}$, where $Z_n = (Z_1, \ldots, Z_i, \ldots, Z_N)$, is quasi-Markovian with memory $M$ if it depends (conditionally) only on the last $M$ steps for the whole process $\{Z_n\}_{n=1}^{\infty}$:*

$$P\left(Z_{i,n} = a_{i,n} \mid \{Z_t = a_t\}_{t=1}^{n-1}\right) = P\left(Z_{i,n} = a_{i,n} \mid \{Z_t = a_t\}_{t=n-M}^{n-1}\right) \quad \textit{for all } n > M.$$

Throughout this work, we will commonly refer to Barabasi-Albert networks [4]. These networks have a random structure that has been shown to resemble the topology of real-world social networks. The underlying graphs are randomly generated sequentially through preferential attachment. An initial complete network is created, and then nodes are added one after the other, each with a fixed number of edges. The new nodes choose existing nodes to connect with at random, but preferentially in the sense that the probability of connecting to an existing node is proportional to its degree $deg(\cdot)$. If there are $k$ existing nodes in the network, the probability that the new node will make an edge to node $i \in \{1, \ldots, k\}$ is $\frac{deg(i)}{\sum_{j=1}^{k} deg(j)}$. In this work, we always start with a 2-node complete network, and our added nodes always create 1 edge. An example of this procedure is shown in Figure 1.2.

## 1.2 Classical Polya Process

We now recall the classical version of the Polya contagion process [19, 38]. The classical Polya process has been applied in many different contexts, including the modelling of communication channels with memory [3], image segmentation [11], as well as in biology, statistics and other areas (see [37]). Consider an urn with $R \in \mathbb{Z}_{>0}$ red balls and $B \in \mathbb{Z}_{>0}$ black balls. We denote the total number of balls by $T$, i.e., $T = R + B$. At each time step, a ball is drawn from the urn. The ball is then returned along with $\Delta \ > 0$ balls of the same colour. We use an indicator $Z_n$ to denote the colour of ball in the $n$th draw:

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

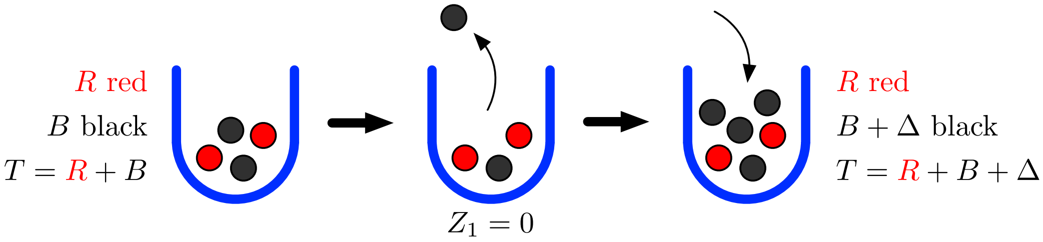


Figure 1.3: Illustration of the first draw for a classical Polya process. We drew a black ball and hence $Z_1 = 0$. Here $R = 2$, $B = 2$, and $\Delta = 2$.

Let $U_n$ denote the proportion of red balls in the urn after the $n$th draw. Then

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

where $\rho_c = \frac{R}{T}$ is the initial proportion of red balls in the urn and $\delta_c = \frac{\Delta}{T}$ is a correlation parameter. We call $\delta_c$ a correlation parameter since the correlation coefficient between any two different draws is constant and of the form

$$Cor(Z_{t_1}, Z_{t_2}) = \frac{\delta_c}{1 + \delta_c} \qquad \text{for all } t_1 \neq t_2 \in \mathbb{Z}_{\geq 1}.$$

Since we draw balls from this urn at each time step, the conditional probability of drawing a red ball at time $n$, given $Z^{n-1} = (Z_1, \cdots, Z_{n-1})$, is given by

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

It can be easily shown that $\{U_n\}_{n=1}^{\infty}$ is a martingale [21]. The process $\{Z_n\}_{n=1}^{\infty}$, whose $n$-fold joint distribution is denoted by $Q_{\rho_c,\delta_c}^{(n)}$, is also exchangeable (hence stationary) and non-ergodic with both $U_n$ and the process sample average $\frac{1}{n}\sum_{i=1}^{n} Z_i$ converging almost surely as $n \to \infty$ to a random variable governed by the Beta distribution with parameters $\frac{\rho_c}{\delta_c}$ and $\frac{1-\rho_c}{\delta_c}$; we denote this probability density function (pdf) by $\mathsf{Beta}(\frac{\rho_c}{\delta_c}, \frac{1-\rho_c}{\delta_c})$ [21, 3]. Lastly, the 1-dimensional distribution of the Polya process is $Q_{\rho_c,\delta_c}^{(1)}(a) = P(Z_n = a) = (\rho_c)^a (1-\rho_c)^{1-a}$, for all $n \in \mathbb{Z}_{\geq 1}$ and $a \in \{0, 1\}$. The above classical Polya process $\{Z_n\}_{n=1}^{\infty}$ is fully described by its parameters $\rho_c$ and $\delta_c$, and thus we denote it by $\mathsf{Polya}(\rho_c, \delta_c)$.

A number of adaptations to the classical Polya process exist in the literature. The most common change is to alter the way that balls are drawn and added. To express the number of balls of each colour that are added after each draw, a replacement matrix $M_R$ is commonly used,

$$M_R = \begin{bmatrix} \Delta_{rr} & \Delta_{rb} \\ \Delta_{br} & \Delta_{bb} \end{bmatrix},$$

where $\Delta_{rr}$ red balls and $\Delta_{rb}$ black balls are added to the urn when a red ball is drawn, and similarly $\Delta_{br}$ red and $\Delta_{bb}$ black balls are added after black is drawn. If $\Delta_{rr} + \Delta_{rb} = \Delta_{br} + \Delta_{bb}$, the process is called *balanced*. Most existing results in the literature are concerned with balanced processes, as it is simpler mathematically; nevertheless, a number of works have investigated properties and limiting behaviours for scenarios with unbalanced urns [41], and even when the parameters are any non-negative integers [32]. Some works consider cases where a random number of balls are drawn, some when a random number are added [7], or both [6]. In many cases the limiting behaviours of these altered processes are considered. In the case where multiple balls are drawn, both with and without replacement, [16] studies the expectation and variance of the number of balls of each colour in time. [28] investigates a model with an arbitrary number of colours of balls, as well as constant factors that alter the probabilities of selecting specific colours, and prove a functional limit theorem for this process. Lastly, and perhaps most influentially, [36] explores the idea that the number of balls added at each step can vary in time.

Further adaptations considered in the literature include allowing the probability of drawing balls to be altered, as well as embeddings of the process in continuous time.

In [26], different functions that depend on the proportion of balls in the urn are used to calculate the probability of drawing one of the two colours. [47] further generalizes this by allowing the functions to be nonlinear, and shows some asymptotic behaviours of the resulting process along with applications. Some authors consider embedding the classical Polya process in continuous time [9], and further works examine its limiting behaviour [10]. Herein we do not consider such adaptations, since they add a large amount of complexity to the analysis without adding a significant benefit.

Finally, one of the most important adaptations that we consider herein is the use of more than one urn. In [15], a model with two urns that influence only one another and not themselves is investigated. [5] presents a model with multiple urns, wherein at each step a single urn is randomly selected and sampled from based on a convex combination of all urn proportions. In the setting of a network, [13] gives each node an urn with balls of one colour. Balls are added based on where edges exist: a ball is randomly added to one of the nodes on the edge based on the proportion of balls in its urn relative to the sum of balls in both nodes' urns. In this way individual interactions between nodes may affect the state of one another. However, all of these models fail to capture the notion of spatial impact between many nodes in a neighbourhood simultaneously. It is with this in mind that we introduce the network Polya contagion process and examine its properties in the next chapter.

# Chapter 2

# Network Polya Contagion Process

In this section, we introduce a generalization of the Polya contagion process to networks, where each individual node in the underlying graph that describes the network topology is still equipped with an urn; however, the node's neighbouring structure affects the evolution of its process. This model hence captures spatial contagion, since infected neighbours increase the chance of a node being infected in the future.

Consider an undirected graph $\mathcal{G} = (V, \mathcal{E})$, where $V = \{1, \ldots, N\}$ is the set of $N \in \mathbb{Z}_{\geq 1}$ nodes and $\mathcal{E} \subset V \times V$ is the set of edges. We assume that $\mathcal{G}$ is connected, i.e., there is a path between any two nodes in $\mathcal{G}$. We use $\mathcal{N}_i$ to denote the set of nodes that are neighbours to node $i$, that is $\mathcal{N}_i = \{v \in V : (i, v) \in \mathcal{E}\}$, and $\mathcal{N}'_i = \{i\} \cup \mathcal{N}_i$. If $\mathcal{N}'_i = V$ for all $i \in V$, the network is called complete; if $|\mathcal{N}_i| = |\mathcal{N}_j|$ for all $i, j \in V$, we call it regular. Each node $i \in V$ is equipped with an urn, initially with $R_i \in \mathbb{Z}_{>0}$ red balls and $B_i \in \mathbb{Z}_{>0}$ black balls (we do not let $R_i = 0$ or $B_i = 0$ to avoid any degenerate cases). We let $T_i = R_i + B_i$ be the total number of balls in the $i$th urn, $i \in \{1, \cdots, N\}$. Thus in the case of the network Polya contagion process, the initial proportion of red balls for the entire network is

$$\rho = \frac{\sum_{i=1}^{N} R_i}{\sum_{i=1}^{N} T_i}. \tag{2.1}$$

We use $Z_{i,n}$ as an indicator for the ball drawn for node $i$ at time $n$:

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

However, instead of drawing solely from its own urn, each node simultaneously draws from their own “super urn”, which is created by combining all the balls in its own urn with the balls in its neighbours’ urns; see Figure 2.1. Hence the super urn of node $i$ initially has $\bar{R}_i = \sum_{j \in \mathcal{N}_i'} R_j$ red balls, $\bar{B}_i = \sum_{j \in \mathcal{N}_i'} B_j$ black balls, and $\bar{T}_i = \sum_{j \in \mathcal{N}_i'} T_j$ balls in total. This construction allows the spatial relationships between nodes to influence their state. This means that $Z_{i,n}$ is the indicator for a ball drawn from node $i$’s super urn, and not its individual urn. Since these variables describe the entire evolution of the process, we call $\{Z_n\}_{n=1}^{\infty} = \{(Z_{1,n}, \ldots, Z_{N,n})\}_{n=1}^{\infty}$ the history of draws, and the natural filtration $\{\mathcal{F}_n\}_{n=1}^{\infty}$ is always assumed to be the natural filtration on $\{Z_n\}_{n=1}^{\infty}$.

We further consider an unbalanced and time-varying version of the classical Polya contagion process, following [36], where the number of balls added after a draw may vary based on the colour that was drawn, the node $i$ that it was drawn for, as well as the time $n$ at which it was drawn. Hence for an arbitrary node $i$, the replacement matrix at time $t$ is

$$M_{R,i}(t) = \begin{bmatrix} \Delta_{r,i}(t) & 0 \\ 0 & \Delta_{b,i}(t) \end{bmatrix}.$$

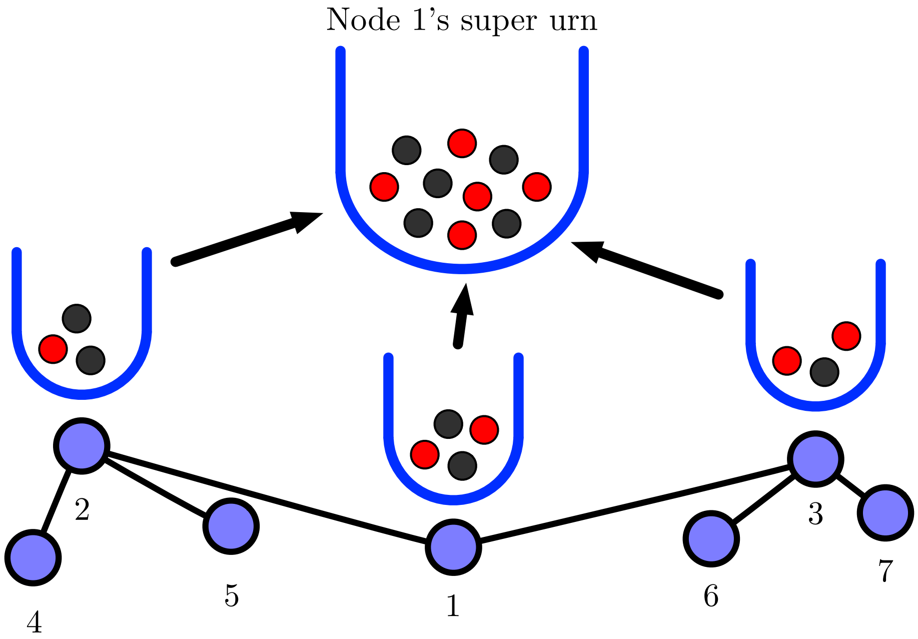


Figure 2.1: Illustration of a super urn in a network.

When $\Delta_{r,i}(t) = \Delta_{b,i}(t)$ for all $t \in \mathbb{Z}_{\geq 1}$, we write $\Delta_i(t)$ instead; if the $\Delta$'s are not node-dependent, we omit the node index. We assume throughout that $\Delta_{r,i}(t) \geq 0, \Delta_{b,i}(t) \geq 0$, for all $t \in \mathbb{Z}_{\geq 1}$ and that there exists $i \in V$ and $t$ such that $\Delta_{r,i}(t) + \Delta_{b,i}(t) \neq 0$; otherwise we are simply sampling with replacement.

In the context of epidemics, the red and black balls in an urn, respectively, represent units of "infection" and "healthiness"; for example, bacteria and white blood cells. In a super urn, the bacteria can infect others in the area and the white blood cells contribute to the overall health in the neighbourhood of an individual.[1] Drawing red at time $t$ means the bacteria in the neighbourhood were successful in reproduction and so the individual was more infected, otherwise they were healthier since the white blood cells reproduced. Thus when $Z_{i,n} = 1$ we say that node $i$ is *infected* at time $n$,

[1] Note that in this context the neighbourhood consists of both the individual $i$ as well as all of its neighbours, which above we define as $\mathcal{N}'_i$.

and if $Z_{i,n} = 0$ then it is *healthy*. We add more units of bacteria once they reproduce, but commonly assume this number, $\Delta_{r,i}(t)$, is the same across all individuals and time because the bacteria does not evolve or become altered. The amount of white blood cells created, $\Delta_{b,i}(t)$, may change since we can give more medicine to certain people to increase their immune response, or vaccinate them so they are better able to fight the disease. However, the application of this model to biological disease is limited by the symmetry of the actions. Traditionally, infection can be shared by individuals through contact, but curing and medicine is limited to the individual; for the network Polya contagion process, both infection and healthiness are shared with all neighbours through the super urns. This symmetry and discrete units of infection and healthiness is justifiable in the context of the spread of ideas and opinions through notions of news articles or social media posts, for computer viruses through infected files and updates to virus definitions, or for advertising between competitors, but requires careful consideration for biological diseases.

A practical adaptation is for urns to have “finite memory” in the sense that the balls added after each draw are only kept in each node’s urn for a finite number $M$ of future draws. The results presented herein will focus on the case where the process has infinite memory unless stated otherwise.

To express the proportion of red balls in the individual urns of the nodes, we define the random vector $U_n = (U_{1,n}, \ldots, U_{N,n})$, where $U_{i,n}$ is the proportion of red balls in node $i$’s urn after the $n$th draw, $i \in V$. For node $i$,

$$U_{i,n} = \frac{R_i + \sum_{t=1}^{n} Z_{i,t}\Delta_{r,i}(t)}{T_i + \sum_{t=1}^{n} Z_{i,t}\Delta_{r,i}(t) + (1 - Z_{i,t})\Delta_{b,i}(t)}$$

where the numerator represents the total number of red balls in node $i$’s urn after the $n$th draw, while the denominator is the total number of balls in the same urn.

Note that $U_{i,0} = \frac{R_i}{T_i}$ is the initial proportion of red balls in node $i$'s urn. For ease of notation, let

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

Furthermore, we define the random vector $S_n = (S_{1,n}, ..., S_{N,n})$ as the proportion of red balls in the super urns of the nodes after the $n$th draw, so that $S_{i,n}$ is the proportion of red balls in node $i$'s super urn after $n$ draws. Hence, for node $i$,

$$S_{i,n} = \frac{\bar{R}_i + \sum_{j\in\mathcal{N}_i'} \sum_{t=1}^{n} Z_{j,t}\Delta_{r,j}(t)}{\sum_{j\in\mathcal{N}_i'} X_{j,n}} = \frac{\sum_{j\in\mathcal{N}_i'} U_{j,n}X_{j,n}}{\sum_{j\in\mathcal{N}_i'} X_{j,n}}, \tag{2.3}$$

where $S_{i,0} = \frac{\bar{R}_i}{\bar{T}_i}$. In fact, $S_{i,n}$ is a function of the random draw variables of the network, and in particular of $\{Z_j^n\}_{j\in\mathcal{N}_i'}$, but for ease of notation, when the arguments are clear, we write $S_{i,n}(Z_1^n, \cdots, Z_N^n) = S_{i,n}$. Then the conditional probability of drawing a red ball from the super urn of node $i$ at time $n$ given the complete network history, i.e. given all the past $n-1$ draw variables for each node in the network $\{Z_j^{n-1}\}_{j=1}^N = \{(Z_{1,1}, \cdots, Z_{1,n-1}), \cdots, (Z_{N,1}, \cdots, Z_{N,n-1})\}$, satisfies

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

That is, the conditional probability of drawing a red ball for node $i$ at time $n$ given the entire past $\{Z_j^{n-1}\}_{j=1}^N$ is the proportion of red balls in its super urn, $S_{i,n-1}$. Since all draws occur simultaneously, this conditional probability for a specific node does not depend on any other draws at time $n$, and hence the draws are conditionally independent. This is analogous to the original Polya case, but instead of relying on the individual proportion of red balls $U_n$ to describe the conditional probability of drawing red balls, we use the super urn proportion of red balls since we now draw from there.

A main objective throughout the rest of this thesis is to study the evolution and stochastic properties of the process. Using the above conditional probability, we can determine the $n$-fold joint probability of the entire network $\mathcal{G}$: for $a_i^n \in \{0,1\}^n$, $i \in \{1, ..., N\}$, we have that

$$\begin{aligned} P_{\mathcal{G}}^{(n)}(a_1^n, \cdots, a_N^n) &:= P\left(\{Z_i^n = a_i^n\}_{i=1}^N\right) \\ &= \prod_{t=1}^{n} P\left(\{Z_{i,t} = a_{i,t}\}_{i=1}^N \mid \{Z_i^{t-1} = a_i^{t-1}\}_{i=1}^N\right) \\ &= \prod_{t=1}^{n}\prod_{i=1}^{N} \Big(S_{i,t-1}\Big)^{a_{i,t}} \Big(1 - S_{i,t-1}\Big)^{1-a_{i,t}}, \end{aligned} \tag{2.5}$$

where $S_{i,t} = S_{i,t}(a_1^t, \cdots, a_N^t)$ is defined in (2.4). Similar to the classical Polya urn process, we are interested in studying the asymptotic behaviour of each node's contagion process $\{Z_{i,n}\}_{n=1}^{\infty}, i \in V$, since understanding many interesting questions regarding the limiting behaviour of epidemics on networks and formulating curing strategies are closely related to this problem. With the above explicit joint distribution, it is possible to determine the distributions of each node's process. More specifically, using (2.5), the $n$-fold distribution of node $i$'s process at time $t \geq n$ is

$$P_{i,t}^{(n)}(a_{i,t-n+1}, ..., a_{i,t}) := \sum_{\substack{a_i^{t-n} \in \{0,1\}^{t-n} \\ a_j^t \in \{0,1\}^t : j \neq i}} P_{\mathcal{G}}^{(t)}(a_1^t, \cdots, a_N^t).$$

We define the *average infection rate* in the network at time $n$ as

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

Since we say that node $i$ is infected at time $n$ when $Z_{i,n} = 1$, $P(Z_{i,n} = 1)$ is the marginal probability that node $i$ is infected at time $n$. Hence to see the marginal probability that an arbitrary node is infected at time $n$, we take the network-wide average. Thus if $\tilde{I}_n$ is high, the probability that an arbitrary node is infected is high. Note that $\tilde{I}_n$ is a function of the network topology $(V, \mathcal{E})$, the initial placement of balls $R_i$ and $B_i$, the draw processes $\{Z_{i,t}\}_{t=1}^n$, and number of balls added $\{\Delta_{r,i}(t)\}_{t=1}^n$ and $\{\Delta_{b,i}(t)\}_{t=1}^n$ for each node $i \in V$. Unfortunately for an arbitrary network, the above quantity does not yield an exact analytical formula (except in the simple case of complete networks). As such, in general it is hard to mathematically analyze the asymptotic behaviour of $\tilde{I}_n$, which we wish to minimize when attempting to cure an epidemic. Instead we examine the asymptotic stochastic behaviour of two closely related variables given by the average individual proportion of red balls at time $n$, namely

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

which we call the *network susceptibility*, and the average neighbourhood proportion of red balls at time $n$,

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

which we call the *network exposure*. Through Equation (2.3) we see that if $U_{i,n}$ increases then this node-specific value causes $S_{j,n}$ to increase for every neighbour $j$ of node $i$, and hence by equation (2.4) their conditional probabilities of drawing red balls increase. More specifically,

$$\uparrow U_{i,n} \quad \overset{(2.3)}{\Longrightarrow} \quad \uparrow S_{j,n} \text{ for all } j \in \mathcal{N}_i' \quad \overset{(2.4)}{\Longrightarrow} \quad \uparrow P\left(Z_{i,n+1} = 1 | \{Z_j^n\}_{j=1}^N\right) \text{ for all } j \in \mathcal{N}_i'.$$

Thus if $\tilde{U}_n$ is high, then this average measure of individual nodes implies that the conditional probability of a node being infected is higher on average. Hence $\tilde{U}_n$ can be understood as the average node prevalence of infection. The effect of the network exposure on this conditional probability is more direct, since Equation (2.4) shows that $\tilde{S}_n$ is in fact the network-wide average of the conditional probability of infection, which is a quantity that is intimately related to the state of infection in the neighbourhood of node $i$. Thus $\tilde{S}_n$ represents the average neighbourhood prevalence of infection. Note that similarly to $\tilde{I}_n$, both $\tilde{U}_n$ and $\tilde{S}_n$ are functions of the network variables.

## 2.1 Stochastic Properties

We next examine the stochastic properties of the network contagion process. We assume throughout the beginning of this section that $\Delta_{r,i}(t) = \Delta_{b,i}(t) = \Delta > 0$, for all $i \in V$ and times $t$; that is the net number of red and black balls added are equal and constant in time for all nodes. In the case of a complete network, the composition of every nodes' super urn is identical, since there is only one super urn that is being drawn from. Thus for a complete network the super urn model is analogous to one urn where multiple draws occur with replacement, which has been recently studied in detail [16]. However, the analysis in [16] is carried out in an aggregate sense, i.e., only for the entire urn and not individual processes. Unfortunately, this aggregate approach does not work in a network setting, whereas the super urn model proposed here is applicable.

### 2.1.1 Complete Network Marginal Distributions

We first focus on the special case of complete networks to derive some useful probability distributions; later on, we will obtain other stochastic properties that apply to more general networks. Given that the network is complete, we focus on one of the nodes, say $i \in V$. For ease of notation, we define $\bar{T}_j = \sum_{k=1}^{N} T_k := \bar{T}$, and similarly, $\bar{R}_j =: \bar{R}, \bar{B}_j =: \bar{B}$, for all $j \in V$. Defining the events $A_{n-1} = \{Z_{i,n-1} = a_{n-1}, ..., Z_{i,1} = a_1\}$ and $W_{n-1}\left(\{b_j^{n-1}\}_{j\neq i}\right) = \{A_{n-1}, \{Z_j^{n-1} = b_j^{n-1}\}_{j\neq i}\}$ with $b_j^{n-1} \in \{0,1\}^{n-1}$, and parameters $\rho = \frac{\bar{R}}{\bar{T}}$ and $\delta = \frac{N\Delta}{\bar{T}}$, using (2.4) under the above assumptions we have

$$\begin{aligned}
&P(Z_{i,n} = 1, A_{n-1}) \\
&= \sum_{b_j^{n-1} \in \{0,1\}^{n-1} : j \neq i} P\left(Z_{i,n} = 1 | W_{n-1}\left(\{b_j^{n-1}\}_{j\neq i}\right)\right) P\left(W_{n-1}\left(\{b_j^{n-1}\}_{j\neq i}\right)\right) \\
&= \sum_{b_j^{n-1} : j \neq i} \frac{\bar{R} + \Delta \sum_{t=1}^{n-1}\left(a_t + \sum_{j\neq i} b_{j,t}\right)}{\bar{T} + \sum_{t=1}^{n-1}\left(\Delta + \sum_{j\neq i}\Delta\right)} P\left(W_{n-1}\left(\{b_j^{n-1}\}_{j\neq i}\right)\right) \\
&= \sum_{b_j^{n-1} : j \neq i} \frac{\rho + \frac{\delta}{N} \sum_{t=1}^{n-1}\left(a_t + \sum_{j\neq i} b_{j,t}\right)}{1 + (n-1)\delta} P\left(W_{n-1}\left(\{b_j^{n-1}\}_{j\neq i}\right)\right) \\
&= \sum_{b_j^{n-1} : j \neq i} \left[\rho \frac{P\left(W_{n-1}\left(\{b_j^{n-1}\}_{j\neq i}\right)\right)}{1+(n-1)\delta} + \frac{\delta}{N} \sum_{t=1}^{n-1} \left( a_t \frac{P\left(W_{n-1}\left(\{b_j^{n-1}\}_{j\neq i}\right)\right)}{1+(n-1)\delta} \right.\right. \\
&\qquad \left.\left. + \sum_{j\neq i} \frac{b_{j,t} P\left(W_{n-1}\left(\{b_j^{n-1}\}_{j\neq i}\right)\right)}{1+(n-1)\delta} \right)\right]. \qquad (2.6)
\end{aligned}$$

By examining an arbitrary term $k \neq i$ in the final sum above, for fixed $t \in \{1, ..., n-1\}$, we can sum out all the other draw variables:

$$\begin{aligned}
\sum_{b_j^{n-1} \in \{0,1\}^{n-1} : j \neq i} b_{k,t} P\left(W_{n-1}\left(\{b_j^{n-1}\}_{j\neq i}\right)\right) &= \sum_{b_j^{n-1} \in \{0,1\}^{n-1} : j \neq i} b_{k,t} P(A_{n-1}, \{Z_j^{n-1} = b_j^{n-1}\}_{j\neq i}) \\
&= \sum_{b_k^{n-1} \in \{0,1\}^{n-1}} b_{k,t} P(A_{n-1}, Z_k^{n-1} = b_k^{n-1}) \\
&= \sum_{b_{k,t} \in \{0,1\}} b_{k,t} P(A_{n-1}, Z_{k,t} = b_{k,t}) \\
&= P(A_{n-1}, Z_{k,t} = 1). \qquad (2.7)
\end{aligned}$$

Further, by the law of total probability,

$$\sum_{b_j^{n-1} \in \{0,1\}^{n-1} : j \neq i} P(A_{n-1}, \{Z_j^{n-1} = b_j^{n-1}\}_{j\neq i}) = P(A_{n-1}). \qquad (2.8)$$

So using (2.7) and (2.8), (2.6) becomes

$$\frac{\rho P(A_{n-1}) + \frac{\delta}{N}\sum_{t=1}^{n-1}\left[a_t P(A_{n-1}) + \sum_{j\neq i} P(A_{n-1}, Z_{j,t}=1)\right]}{1+(n-1)\delta}.$$

Thus, using the law of total probability, we have

$$\begin{aligned}
&P(Z_{i,n}=1)\\
&= \sum_{a^{n-1}\in\{0,1\}^{n-1}} P(Z_{i,n}=1, A_{n-1})\\
&= \sum_{a^{n-1}}\left[\frac{\rho P(A_{n-1}) + \frac{\delta}{N}\sum_{t=1}^{n-1} a_t P(A_{n-1})}{1+(n-1)\delta} + \frac{\frac{\delta}{N}\sum_{t=1}^{n-1}\sum_{j\neq i} P(A_{n-1}, Z_{j,t}=1)}{1+(n-1)\delta}\right]\\
&= \frac{\rho + \frac{\delta}{N}\sum_{t=1}^{n-1}\sum_{j=1}^{N} P(Z_{j,t}=1)}{1+(n-1)\delta}. \qquad (2.9)
\end{aligned}$$

An interesting corollary of this derivation is as follows.

**Lemma 2.1.1. (Complete Network Marginal Distribution):** *The one dimensional marginal distribution of node i's contagion draw process $\{Z_{i,n}\}_{n=1}^{\infty}$ for the N-node complete network is given by*

$$P_{i,n}^{(1)} = P(Z_{i,n}=a) = \rho^a(1-\rho)^{1-a},$$

*where $i \in V$, $n \geq 1$, and $a \in \{0,1\}$.*

*Proof.* We proceed using strong induction on $n \geq 1$, showing that $P(Z_{i,n}=1)=\rho$, for all nodes $i \in V$ and all $n$. The base case readily holds, since at time $n=1$,

$$P(Z_{1,1}=1) = \cdots = P(Z_{N,1}=1) = \frac{\sum_{i=1}^{N} R_i}{\sum_{i=1}^{N} T_i} = \rho.$$

Now, assuming that $P(Z_{j,t}=1)=\rho$ for all $j \in V$ and $t \leq n$ and using (2.9), we have

$$\begin{aligned}
P(Z_{i,n+1}=1) &= \frac{\rho + \frac{\delta}{N}\sum_{t=1}^{n}\sum_{j=1}^{N} P(Z_{j,t}=1)}{1+n\delta}\\
&= \frac{\rho + \sum_{t=1}^{n}\frac{\delta}{N}N\rho}{1+n\delta}
\end{aligned}$$

$$= \frac{\rho + \delta \sum_{t=1}^{n} \rho}{1 + n\delta} = \rho,$$

which completes the induction argument. The result now follows using the fact that

$$P(Z_{j,n} = 1) + P(Z_{j,n} = 0) = 1 \Rightarrow P(Z_{j,n} = 0) = 1 - \rho,$$

for all $j \in V$ and all $n$. □

This result is analogous to the result we saw for the classical Polya process, and implies that the network Polya contagion process is stationary in its 1-fold distribution. In fact, it is possible to show that our process exhibits stationarity behaviour in some higher-order distributions as well. In our next result, we will see that there is a consistent relationship between the draws at the 1st and $n$th time steps.

**Lemma 2.1.2. (Complete Network $(n, 1)$-step Marginal Probability):** *For the complete network, the 2-dimensional marginal probability that node $i$'s draw variables at times $n$ and $1$ are both one is given by*

$$P(Z_{i,n} = 1, Z_{i,1} = 1) = \rho \frac{\rho + (1 + (N-1)\rho)\frac{\delta}{N}}{1 + \delta},$$

*for $i \in V$, $n \geq 2$. Furthermore, for any other node $k$,*

$$P(Z_{k,n} = 1, Z_{i,1} = 1) = \rho \frac{\rho + (1 + (N-1)\rho)\frac{\delta}{N}}{1 + \delta}.$$

*Proof.* By Lemma 2.1.1 we have that $P(Z_{k,1} = 1) = \rho$ for all $k \in V$, so it is enough to show that

$$P(Z_{k,n} = 1 \mid Z_{i,1} = 1) = \frac{\rho + (1 + (N-1)\rho)\frac{\delta}{N}}{1 + \delta} \tag{2.10}$$

for all $n$ and nodes $i$ and $k$. Using the law of total probability, (2.4), and after some simplifications, with $W_{n-1}\left(a_2^{n-1}, \{b_j^{n-1}\}_{j \neq i}\right) := \{Z_{i,2}^{n-1} = a_2^{n-1}, \{Z_{j,1}^{n-1} = b_{j,1}^{n-1}\}_{j \neq i}\}$, we have that

$$\begin{aligned}
&P(Z_{i,n}=1 \mid Z_{i,1}=1)\\
&= \sum_{\substack{a_2^{n-1}\in\{0,1\}^{n-2}\\ b_{j,1}^{n-1}\in\{0,1\}^{n-1}:j\neq i}} P\left(Z_{i,n}=1 \mid Z_{i,1}=1, W_{n-1}\left(a_2^{n-1},\{b_j^{n-1}\}_{j\neq i}\right)\right)\\
&\qquad\qquad \times P\left(W_{n-1}\left(a_2^{n-1},\{b_j^{n-1}\}_{j\neq i}\right) \mid Z_{i,1}=1\right)\\
&= \sum_{a_2^{n-1},b_{j,1}^{n-1}:j\neq i} \frac{\rho+\frac{\delta}{N}(1+\sum_{t=2}^{n-1}a_t+\sum_{t=1}^{n-1}\sum_{j\neq i}b_{j,t})}{1+(n-1)\delta} P\left(W_{n-1}\left(a_2^{n-1},\{b_j^{n-1}\}_{j\neq i}\right) \mid Z_{i,1}=1\right)\\
&= \sum_{a_2^{n-1},b_{j,1}^{n-1}:j\neq i} \frac{P\left(W_{n-1}\left(a_2^{n-1},\{b_j^{n-1}\}_{j\neq i}\right) \mid Z_{i,1}=1\right)}{1+(n-1)\delta}\left[\left(\rho+\frac{\delta}{N}\right)+\frac{\delta}{N}\sum_{t=2}^{n-1}a_t+\frac{\delta}{N}\sum_{t=1}^{n-1}b_{j,t}\right].
\end{aligned}$$

Then, after arranging terms and using the law of total probability for

$$\sum_{a_2^{n-1},b_{j,1}^{n-1}:j\neq i} b_{j,t}P\left(W_{n-1}\left(a_2^{n-1},\{b_j^{n-1}\}_{j\neq i}\right) \mid Z_{i,1}=1\right)=P(Z_{j,t}=1 \mid Z_{i,1}=1),$$

we have

$$\begin{aligned}
&P(Z_{i,n}=1 \mid Z_{i,1}=1)\\
&= \frac{(\rho+\frac{\delta}{N})(1)}{1+(n-1)\delta}+\frac{\frac{\delta}{N}\sum_{t=2}^{n-1}P(Z_{i,t}=1|Z_{i,1}=1)}{1+(n-1)\delta}+\frac{\frac{\delta}{N}\sum_{t=1}^{n-1}\sum_{j\neq i}P(Z_{j,t}=1|Z_{i,1}=1)}{1+(n-1)\delta}\\
&= \frac{\rho+\frac{\delta}{N}\sum_{j\neq i}P(Z_{j,1}=1)}{1+(n-1)\delta}+\frac{\frac{\delta}{N}\left[1+\sum_{j=1}^{N}\sum_{t=2}^{n-1}P(Z_{j,t}=1|Z_{i,1}=1)\right]}{1+(n-1)\delta}\\
&= \frac{\rho(1+(N-1)\frac{\delta}{N})}{1+(n-1)\delta}+\frac{\frac{\delta}{N}\left[1+\sum_{j=1}^{N}\sum_{t=2}^{n-1}P(Z_{j,t}=1 \mid Z_{i,1}=1)\right]}{1+(n-1)\delta}. \qquad (2.11)
\end{aligned}$$

It can be similarly shown by symmetry of the complete network that (2.11) holds for $P(Z_{k,n}=1 \mid Z_{i,1}=1)$ if $k\neq i$.

In order to show (2.10), we proceed using strong induction on $n\geq 2$. For the base case, setting $n=2$ in (2.11), we have for any $i,k\in V$,

$$P(Z_{k,2}=1|Z_{i,1}=1)=\frac{\rho(1+(N-1)\frac{\delta}{N})+\frac{\delta}{N}}{1+\delta}=\frac{\rho+(1+(N-1)\rho)\frac{\delta}{N}}{1+\delta}.$$

as desired. Assume now that $P(Z_{k,t} = 1 \mid Z_{i,1} = 1)$ is given by (2.10), for $2 \le t \le n-1$ and any $i, k \in V$. Then by (2.11),

$$
\begin{aligned}
&P(Z_{k,n} = 1 \mid Z_{i,1} = 1) \\
&= \frac{\rho(1 + (N-1)\frac{\delta}{N})}{1 + (n-1)\delta} + \frac{\frac{\delta}{N}\left[1 + \sum_{j=1}^{N} \sum_{t=2}^{n-1} P(Z_{j,t} = 1 \mid Z_{i,1} = 1)\right]}{1 + (n-1)\delta} \\
&= \frac{\rho(1 + (N-1)\frac{\delta}{N}) + \frac{\delta}{N}\left[N(n-2)\frac{\rho + (1+(N-1)\rho)\frac{\delta}{N}}{1+\delta}\right]}{1 + (n-1)\delta} \\
&= \frac{1}{1 + (n-1)\delta}\left[(1+\delta)\frac{\rho + (1 + (N-1)\rho)\frac{\delta}{N}}{1+\delta} + \delta(n-2)\frac{\rho + (1 + (N-1)\rho)\frac{\delta}{N}}{1+\delta}\right] \\
&= \frac{\rho + (1 + (N-1)\rho)\frac{\delta}{N}}{1+\delta} \times \frac{(1+\delta) + \delta(n-2)}{1 + (n-1)\delta} \\
&= \frac{\rho + (1 + (N-1)\rho)\frac{\delta}{N}}{1+\delta},
\end{aligned}
$$

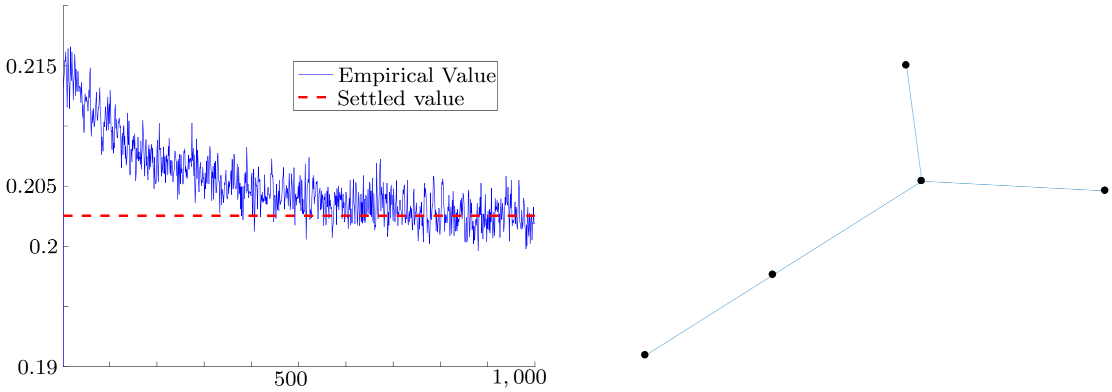
which completes the induction argument. □

While these two results suggest that the network Polya contagion process is stationary, we may show that each node's draw process is not stationary in general, and hence is different from the classical $\mathsf{Polya}(\rho_c, \delta_c)$ process.

**Remark 2.1.3. (Non-Stationarity of the Network Contagion Process):** *Consider a 2-node complete network. Then, using* (2.5), *one can obtain (after some simplifications) that*

$$
\begin{aligned}
P(Z_{1,2} = 1, Z_{1,1} = 1) &= \rho\frac{\rho + (1+\rho)\frac{\delta}{2}}{1+\delta}, \\
P(Z_{1,3} = 1, Z_{1,2} = 1) &= \sum_{\substack{a_1 \in \{0,1\} \\ b^3 \in \{0,1\}^3}} P(Z_{1,1} = a_1, \{Z_{1,t} = 1\}_{t=2}^{3}, \{Z_{2,t} = b_t\}_{t=1}^{3}) \\
&= \rho\frac{4\rho + \delta(2 + 14\rho) + \delta^2(6 + 14\rho) + \delta^3(5 + 3\rho)}{4(1+\delta)^2(1+2\delta)},
\end{aligned}
$$

*and hence the network process is not stationary.* •



(a) Plot of $P(Z_{i,n} = 1, Z_{i,n-1} = 1)$

(b) 5-node Barabasi-Albert network [4]

Figure 2.2: Simulated values for $P(Z_{i,n} = 1, Z_{i,n-1} = 1)$ for an arbitrary node $i$ averaged over 50,000 simulated trials. Here we observe asymptotic stationarity, as for some large enough $n$ deviations from the settled value are very small in magnitude.

Since every exchangeable process is necessarily stationary, Remark 2.1.3 implies that the network Polya process is also not exchangeable in general. Although this is the case, simulated results suggest that it satisfies some asymptotic stationarity properties. While precise definitions for asymptotic stationarity exist [45], here we simply mean that given sufficient time, the joint probability distribution of the process settles. A representative example of this phenomenon is shown in Figure 2.2 for the 2-dimensional distribution at times $n$ and $n-1$ in a 5-node network. This phenomenon along with the results of Lemmas 2.1.1 and 2.1.2 motivates the use of the classical Polya process, which we know is stationary, to approximate the draw process of a single node, as seen in Section 2.3.

### 2.1.2 Finite Memory

In some applications it makes sense for our process to have finite memory. The idea is that balls that are added to urns remain only for a finite period of time, say $M$ steps. After this set amount of time, balls are removed from the urn. Hence, after the draw has occurred at time $t$ and we have added the new balls, we would remove $\Delta_{r,i}(t-M)$ red balls or $\Delta_{b,i}(t-M)$ black balls from node $i$'s urn, depending on what was added $M$ steps ago. This model is developed in [3] for the classical Polya process in the context of modelling communication channels, where it is shown that the resulting finite memory contagion process is stationary, Markovian and ergodic. We now prove that we have similar results for the network Polya contagion process: the individual processes are in fact quasi-Markovian, since they depend only on what has occurred during the last $M$ steps for the entire process, and the entire state is Markovian with memory $M$.

**Theorem 2.1.4. (Finite Memory Markovity):** *For the network Polya contagion process with finite memory $M$, the individual draw processes $\{Z_{i,n}\}_{n=1}^{\infty}$ are quasi-Markovian with memory $M$ for all $i \in V$. Further, the draw process for the entire network $\{Z_n\}_{n=1}^{\infty}$, where $Z_n = (Z_{1,n}, \ldots, Z_{N,n})$, is Markovian with memory $M$.*

*Proof.* First we will focus on an arbitrary node $i$. By Equations (2.2) and (2.4) and the fact that added balls are removed after $M$ steps, we have for $n > M$ that

$$\begin{aligned}
&P\left(Z_{i,n} = 1 \mid \{Z_j^{n-1}\}_{j=1}^{N}\right) \\
&= \frac{\bar{R}_i + \sum_{j \in \mathcal{N}_i'} \left( \sum_{t=1}^{n-1} Z_{j,t}\Delta_{r,j}(t) - \sum_{t=1}^{n-M-1} Z_{j,t}\Delta_{r,j}(t) \right)}{\sum_{j \in \mathcal{N}_i'} X_{j,n-1} - X_{j,n-M-1}} \\
&= \frac{\bar{R}_i + \sum_{j \in \mathcal{N}_i'} \sum_{t=n-M}^{n-1} Z_{j,t}\Delta_{r,j}(t)}{\bar{T}_i + \sum_{j \in \mathcal{N}_i'} \sum_{t=n-M}^{n-1} Z_{j,t}\Delta_{r,j}(t) + (1 - Z_{j,t})\Delta_{b,j}(t)} \\
&= P\left(Z_{i,n} = 1 \mid \{Z_{j,n-M}^{n-1}\}_{j=1}^{N}\right).
\end{aligned}$$

Hence the individual draw process is quasi-Markovian in that it only cares about the last $M$ draws that have occurred in the whole network. Now we can examine the entire draw process. So using the result above along with conditional independence of the draws, for $a = (a_1, \dots, a_N) \in \{0,1\}^N$ we have for $n > M$ that

$$\begin{aligned}
&P\left(Z_{1,n} = a_1, \dots, Z_{N,n} = a_N \mid \{Z_j^{n-1}\}_{j=1}^N\right) \\
&= \prod_{i=1}^N P\left(Z_{i,n} = 1 \mid \{Z_j^{n-1}\}_{j=1}^N\right) \\
&= \prod_{i=1}^N P\left(Z_{i,n} = 1 \mid \{Z_{j,n-M}^{n-1}\}_{j=1}^N\right) \\
&= P\left(Z_{1,n} = a_1, \dots, Z_{N,n} = a_N \mid \{Z_{j,n-M}^{n-1}\}_{j=1}^N\right),
\end{aligned}$$

and hence the entire network process $\{Z_n\}_{n=1}^\infty$ is Markovian with memory $M$. □

By the definition of the Markov property, this result suggests that the network Polya contagion process with finite memory is a limited reinforcement process. The effect of a draw is not permanent, and instead only persists for $M$ steps; however, during this time window it still continues to affect the evolution of the process. This behaviour is interesting, and indeed one can study the upcoming subjects within this context. However, herein we focus on the scenario with infinite memory and leave this adaptation as an important future direction.

## 2.2 Martingale Theorems

We now turn our attention to the martingale properties of the network contagion process, where we do not assume that the network is necessarily complete. Recall that by the martingale convergence theorem, if a process $\{Z_n\}_{n=1}^\infty$ is a martingale (or supermartingale, or submartingale), there exists a random variable $Z$ such that $\{Z_n\}_{n=1}^\infty$ converges almost surely to $Z$ as $n \to \infty$.

**Theorem 2.2.1. (Individual Urn Proportion Martingale):** *For a network $\mathcal{G} = (V, \mathcal{E})$, $\Delta_{r,i}(n) = \Delta_{b,i}(n) = \Delta$ and $T_i = T$, for all $i \in V$ and all $n$, the individual proportion of red balls $\{U_{i,n}\}_{n=1}^{\infty}$ is a martingale with respect to the draws for the whole network $\{Z_n\}_{n=1}^{\infty} = \{(Z_{1,n}, ..., Z_{N,n})\}_{n=1}^{\infty}$ if and only if, almost surely,*

$$\frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} U_{j,n-1} = U_{i,n-1}.$$

*Proof.* Using the expression for $U_{i,n}$, (2.3), and (2.4), we have

$$\begin{aligned}
E[U_{i,n} \mid Z_{n-1}] &= E\left[\frac{\Delta Z_{i,n} + U_{i,n-1}(T + (n-1)\Delta)}{T + n\Delta} \;\Big|\; Z_{n-1}\right] \\
&= \frac{U_{i,n-1}(T + (n-1)\Delta)}{T + n\Delta} + \frac{\Delta E[Z_{i,n} \mid Z_{n-1}]}{T + n\Delta} \\
&= U_{i,n-1}\frac{T + (n-1)\Delta}{T + n\Delta} + \frac{\Delta P(Z_{i,n} = 1 | Z^{n-1})}{T + n\Delta} \\
&= U_{i,n-1}\left(1 - \frac{\Delta}{T + n\Delta}\right) + \frac{\Delta \sum_{j \in \mathcal{N}_i'} U_{j,n-1}(T + (n-1)\Delta)}{(T + n\Delta)|\mathcal{N}_i'|(T + (n-1)\Delta)} \\
&= U_{i,n-1} - \frac{\Delta U_{i,n-1}}{T + n\Delta} + \Delta \frac{U_{i,n-1} + \sum_{j \in \mathcal{N}_i} U_{j,n-1}}{|\mathcal{N}_i'|(T + n\Delta)} \\
&= U_{i,n-1} + \Delta \frac{(1 - |\mathcal{N}_i'|)U_{i,n-1} + \sum_{j \in \mathcal{N}_i} U_{j,n-1}}{|\mathcal{N}_i'|(T + n\Delta)} \\
&= U_{i,n-1} + \Delta \frac{\left[\sum_{j \in \mathcal{N}_i} U_{j,n-1}\right] - |\mathcal{N}_i| U_{i,n-1}}{(T + n\Delta)(|\mathcal{N}_i| + 1)} \\
&= U_{i,n-1} + \frac{\Delta \sum_{j \in \mathcal{N}_i}(U_{j,n-1} - U_{i,n-1})}{(T + n\Delta)(|\mathcal{N}_i| + 1)}. \qquad (2.12)
\end{aligned}$$

This implies that $\{U_{i,n}\}_{n=1}^{\infty}$ is a martingale with respect to $\{Z_n\}_{n=1}^{\infty}$ if and only if

$$\sum_{j \in \mathcal{N}_i} U_{j,n-1} - U_{i,n-1} = 0 \Leftrightarrow \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} U_{j,n-1} = U_{i,n-1}.$$

almost surely. □

If the condition in Theorem 2.2.1 holds, we obtain by the martingale convergence theorem [8, 23], that for any $i$, both $U_{i,n}$ and $\frac{1}{n}\sum_{t=1}^{n} Z_{i,t}$ converge almost surely to a limit as $n \to \infty$. However, the condition of Theorem 2.2.1, barring the trivial single node scenario (which reverts to the classical Polya scheme), is not verifiable. To resolve this issue, we instead examine the evolution of the average proportion of red balls (i.e., the susceptibility) in a regular network.

**Theorem 2.2.2. (Regular Network Susceptibility Martingale):** *For a regular network $\mathcal{G} = (V, \mathcal{E})$ with $\Delta_{r,i}(n) = \Delta_{b,i}(n) = \Delta$ and $T_i = T$ for all nodes $i \in V$ and times $n$, the network susceptibility $\{\tilde{U}_n\}_{n=1}^{\infty}$, where $\tilde{U}_n = \frac{1}{N}\sum_{i=1}^{N} U_{i,n}$, is a martingale with respect to $\{Z_n\}_{n=1}^{\infty}$.*

*Proof.* We have, similar to Theorem 2.2.1, that

$$\begin{aligned} E[\tilde{U}_n \mid Z_{n-1}] &= \frac{1}{N}\sum_{i=1}^{N} E[U_{i,n} \mid Z_{n-1}] \\ &= \frac{1}{N}\sum_{i=1}^{N}\left[U_{i,n-1} + \frac{\Delta \sum_{j\in\mathcal{N}_i} U_{j,n-1} - U_{i,n-1}}{(T+n\Delta)(|\mathcal{N}_i|+1)}\right] \\ &= \tilde{U}_{n-1} + \sum_{i=1}^{N} \frac{\Delta \sum_{j\in\mathcal{N}_i} U_{j,n-1} - U_{i,n-1}}{N(T+n\Delta)(|\mathcal{N}_i|+1)}. \end{aligned}$$

Let us examine the second term of the last equality. If this term is zero, we will have that $\{\tilde{U}_n\}_{n=1}^{\infty}$ is a martingale with respect to $\{Z_n\}_{n=1}^{\infty}$. We can rewrite this term by defining the adjacency matrix $[a_{ij}]$ of our network, where the $(i, j)$th entry $a_{ij}$ is 1 if $(i, j) \in \mathcal{E}$, and 0 otherwise. Since we assumed that our network was undirected, this means that $[a_{ij}]$ is symmetric, i.e., $a_{ij} = a_{ji}$ for all $i, j \in V$. So let us rewrite the term above:

$$\begin{aligned} \sum_{i=1}^{N} \frac{\Delta \sum_{j\in\mathcal{N}_i} U_{j,n-1} - U_{i,n-1}}{N(T+n\Delta)(|\mathcal{N}_i|+1)} &= \sum_{i=1}^{N} \frac{\Delta \sum_{j=1}^{N} a_{ij}(U_{j,n-1} - U_{i,n-1})}{N(T+n\Delta)(|\mathcal{N}_i|+1)} \\ &= \frac{\Delta}{N(T+n\Delta)} \sum_{i=1}^{N}\sum_{j=1}^{N} \frac{a_{ij}(U_{j,n-1} - U_{i,n-1})}{|\mathcal{N}_i|+1}. \end{aligned}$$

Now, we examine the sum of the $(i, j)$ and $(j, i)$ components of the double sum, where

$(i,j) \in \mathcal{E}$ (otherwise both terms are zero). Recall that $(i,i) \notin \mathcal{E}$, $\forall i$. We have

$$\begin{aligned}
&\frac{a_{ij}(U_{j,n-1} - U_{i,n-1})}{|\mathcal{N}_i| + 1} + \frac{a_{ji}(U_{i,n-1} - U_{j,n-1})}{|\mathcal{N}_j| + 1} \\
&= \frac{a_{ij}(U_{j,n-1} - U_{i,n-1})(|\mathcal{N}_j| + 1)}{(|\mathcal{N}_i| + 1)(|\mathcal{N}_j| + 1)} + \frac{a_{ji}(U_{i,n-1} - U_{j,n-1})(|\mathcal{N}_i| + 1)}{(|\mathcal{N}_i| + 1)(|\mathcal{N}_j| + 1)} \\
&= \frac{a_{ij}\left[U_{j,n-1}(|\mathcal{N}_j| - |\mathcal{N}_i|) + U_{i,n-1}(|\mathcal{N}_i| - |\mathcal{N}_j|)\right]}{(|\mathcal{N}_i| + 1)(|\mathcal{N}_j| + 1)} \\
&= \frac{a_{ij}(|\mathcal{N}_j| - |\mathcal{N}_i|)}{(|\mathcal{N}_i| + 1)(|\mathcal{N}_j| + 1)}\Big(U_{j,n-1} - U_{i,n-1}\Big).
\end{aligned}$$

From above, it is clear that this term is zero for all $i$ and $j$ by setting $|\mathcal{N}_j| = |\mathcal{N}_i|$, i.e. in any regular network, and so $\{\tilde{U}_n\}_{n=1}^{\infty}$ is a martingale with respect to $\{Z_n\}_{n=1}^{\infty}$. □

We next allow the net number of black balls $\Delta_{b,i}(\cdot)$ to evolve stochastically in time as a function of the past draw history in the network in order to steer $\{U_{i,n}\}_{n=1}^{\infty}$ to a limit for every node $i$. An important assumption used herein is that the number of red balls to be added $\Delta_{r,i}(n)$ is known at least one step ahead of time so that in particular for the natural filtration $\{\mathcal{F}_n\}_{n=1}^{\infty}$ on the entire history of draws $\{Z_n\}_{n=1}^{\infty}$, $\Delta_{r,i}(n)$ is almost surely constant given $\mathcal{F}_{n-1}$. A sufficient, but not necessary, condition to satisfy this assumption is for $\{\Delta_{r,i}(n)\}_{n=1}^{\infty}$ to be set, for all $i \in V$, before the process begins.

**Theorem 2.2.3. (Individual Urn Proportion Categories):** *In a general network $\mathcal{G} = (V, \mathcal{E})$, if we choose $\{\Delta_{b,i}(n)\}_{n=1}^{\infty}$ so that*

$$\Delta_{b,i}(n) \geq \frac{\Delta_{r,i}(n)(1 - U_{i,n-1})S_{i,n-1}}{U_{i,n-1}(1 - S_{i,n-1})}$$

*almost surely for all $n \in \mathbb{Z}_{\geq 1}$ and $i \in V$ (resp. equal to, less than or equal to) then $\{U_{i,n}\}_{n=1}^{\infty}$ is a supermartingale (resp. martingale, submartingale) with respect to the natural filtration $\{\mathcal{F}_n\}_{n=1}^{\infty}$, i.e.,*

$$E[U_{i,n}|\mathcal{F}_{n-1}] \leq U_{i,n-1} \qquad \textit{almost surely } \forall n \in \mathbb{Z}_{\geq 1}.$$

*Proof.* We will start with the case of a supermartingale. That is, we wish to show that almost surely for all $n \in \mathbb{Z}_{\geq 1}$,

$$E[U_{i,n} \mid \mathcal{F}_{n-1}] - U_{i,n-1} \leq 0 \Leftrightarrow E[U_{i,n} - U_{i,n-1} \mid \mathcal{F}_{n-1}] \leq 0,$$

since $U_{i,n-1}$ is almost surely constant given $\mathcal{F}_{n-1}$. Take $X_{i,n}$ as in (2.2). We then compute the difference $U_{i,n} - U_{i,n-1}$,

$$\begin{aligned}
&U_{i,n} - U_{i,n-1} \\
&= \frac{R_i + \sum_{t=1}^{n} \Delta_{r,i}(t) Z_{i,t}}{X_{i,n}} - \frac{R_i + \sum_{t=1}^{n-1} \Delta_{r,i}(t) Z_{i,t}}{X_{i,n-1}} \\
&= \frac{\Delta_{r,i}(n) Z_{i,n}}{X_{i,n}} - \frac{(R_i + \sum_{t=1}^{n-1} \Delta_{r,i}(t) Z_{i,t})(X_{i,n} - X_{i,n-1})}{X_{i,n-1} X_{i,n}} \\
&= \frac{\Delta_{r,i}(n) Z_{i,n} - U_{i,n-1}(X_{i,n} - X_{i,n-1})}{X_{i,n}} \\
&= \frac{\Delta_{r,i}(n) Z_{i,n} - U_{i,n-1}(\Delta_{r,i}(n) Z_{i,n} + \Delta_{b,i}(n)(1 - Z_{i,n}))}{X_{i,n}}.
\end{aligned}$$

Since $X_{i,n} > 0$ almost surely, for all $n \in \mathbb{Z}_{\geq 1}$, it will not change the sign of the inequality later on, and so we can ignore it. Thus we wish to check if, almost surely,

$$E\left[\Delta_{r,i}(n) Z_{i,n} - U_{i,n-1}(\Delta_{r,i}(n) Z_{i,n} + \Delta_{b,i}(n)(1 - Z_{i,n})) | \mathcal{F}_{n-1}\right] \leq 0.$$

Now if

$$\Delta_{b,i}(n) \geq \frac{\Delta_{r,i}(n)(1 - U_{i,n-1}) S_{i,n-1}}{U_{i,n-1}(1 - S_{i,n-1})}$$

almost surely, we have

$$\begin{aligned}
&E\big[\Delta_{r,i}(n) Z_{i,n}(1 - U_{i,n-1}) - U_{i,n-1}(1 - Z_{i,n}) \Delta_{b,i}(n) | \mathcal{F}_{n-1}\big] \\
&\leq E\left[\Delta_{r,i}(n) Z_{i,n}(1 - U_{i,n-1}) - U_{i,n-1}(1 - Z_{i,n}) \frac{\Delta_{r,i}(n)(1 - U_{i,n-1}) S_{i,n-1}}{U_{i,n-1}(1 - S_{i,n-1})} \middle| \mathcal{F}_{n-1}\right] \\
&= \Delta_{r,i}(n)(1 - U_{i,n-1}) \left[S_{i,n-1} - (1 - S_{i,n-1}) \frac{S_{i,n-1}}{1 - S_{i,n-1}}\right] \\
&= 0,
\end{aligned}$$

where the second to last equality comes from the fact that $E[Z_{i,n}|\mathcal{F}_{n-1}] = P(Z_{i,n} = 1|\mathcal{F}_{n-1}) = S_{i,n-1}$ almost surely by (2.4), and that $S_{i,n-1}$ is almost surely constant given $\mathcal{F}_{n-1}$. Thus as long as $\Delta_{b,i}(n)$ obeys this bound almost surely for all $n \in \mathbb{Z}_{\geq 1}$, $\{U_{i,n}\}_{n=1}^{\infty}$ is a supermartingale with respect to $\{Z_n\}_{n=1}^{\infty}$. Similarly, if $\Delta_{b,i}(n)$ is almost surely equal (resp. less than or equal) to this bound, $\{U_{i,n}\}_{n=1}^{\infty}$ is a martingale (resp. submartingale) with respect to $\{\mathcal{F}_n\}_{n=1}^{\infty}$. □

**Corollary 2.2.4. (Network Susceptibility Supermartingale):** *In a general network $\mathcal{G} = (V, \mathcal{E})$, if the curing policies $\{\Delta_{b,i}(t)\}_{t=1}^{\infty}$ obey the bound*

$$\Delta_{b,i}(n) \geq \frac{\Delta_{r,i}(n)(1 - U_{i,n-1})S_{i,n-1}}{U_{i,n-1}(1 - S_{i,n-1})}$$

*almost surely for all nodes $i \in V$, then the network susceptibility $\{\tilde{U}_n\}_{n=1}^{\infty}$, where $\tilde{U}_n = \frac{1}{N}\sum_{i=1}^{N} U_{i,n}$, is a supermartingale with respect to the natural filtration $\{\mathcal{F}_n\}_{n=1}^{\infty}$, i.e.,*

$$E[\tilde{U}_n|\mathcal{F}_{n-1}] \leq \tilde{U}_{n-1} \qquad \text{almost surely } \forall n \in \mathbb{Z}_{\geq 1}.$$

*Proof.* Since $\tilde{U}_n$ is simply the average of the $U_{i,n}$'s for all $i \in V$, which are each supermartingales under our conditions by Theorem 2.2.3, the network susceptibility is itself a supermartingale. □

While Corollary 2.2.4 is useful, the network exposure $\tilde{S}_n$ is more closely related to the average infection rate $\tilde{I}_n$ than the network susceptibility $\tilde{U}_n$, since our draws are taken from the super urn. It is with this in mind that we show the next results, which give us sufficient conditions for $\{S_{i,n}\}_{n=1}^{\infty}$ and $\{\tilde{S}_n\}_{n=1}^{\infty}$ to be supermartingales.

**Theorem 2.2.5. (Super Urn Proportion Supermartingale):** *In a general network $\mathcal{G} = (V, \mathcal{E})$, if the curing policy $\{\Delta_{b,i}(t)\}_{t=1}^{\infty}$ obeys the bound*

$$\Delta_{b,i}(n) > \Delta_{r,i}(n)\frac{S_{i,n-1}}{1 - S_{i,n-1}} \max_{k \text{ s.t. } i \in \mathcal{N}'_k} \frac{1 - S_{k,n-1}}{S_{k,n-1}}$$

*almost surely for all nodes $i \in V$, then the neighbourhood proportions of red balls $\{S_{i,n}\}_{n=1}^{\infty}$ are strict supermartingales with respect to the natural filtration $\{\mathcal{F}_n\}_{n=1}^{\infty}$, i.e.*

$$E[S_{i,n}|\mathcal{F}_{n-1}] < S_{i,n-1} \qquad \textit{almost surely } \forall i \in V, n \in \mathbb{Z}_{\geq 1}.$$

*Proof.* First, note that the question of $\{S_{i,n}\}_{n=1}^{\infty}$ being a strict supermartingale is equivalent to

$$E[S_{i,n}|\mathcal{F}_{n-1}] - S_{i,n-1} < 0$$

where $\{\mathcal{F}_n\}$ is the natural filtration (indeed, we can just condition on $Z^{n-1}$). Note, in particular, that $E[Z_{i,t}|\mathcal{F}_n] = Z_{i,t}$ almost surely, for all $i \in V$ and $t \in \{1, \dots, n\}$, and the same is true for $\{S_{i,t}\}_{t=1}^{n}$. For ease of notation, we let

$$\bar{X}_{i,n} = \bar{T}_i + \sum_{j \in \mathcal{N}'_i} \sum_{t=1}^{n} \Delta_r Z_{j,t} + \Delta_{b,j}(t)(1 - Z_{j,t}) = \sum_{j \in \mathcal{N}'_i} X_{j,t}.$$

Then almost surely, as in Theorem 2.2.3,

$$S_{i,n} - S_{i,n-1} = \frac{S_{i,n-1}(\bar{X}_{i,n-1} - \bar{X}_{i,n}) + \sum_{j \in \mathcal{N}'_i} \Delta_{r,j}(n) Z_{j,n}}{\bar{X}_{i,n}}.$$

Since $\bar{X}_{i,n} > 0$ almost surely for all $n \in \mathbb{Z}_{\geq 1}$ and all $i \in V$, we can ignore it. Further, since $S_{i,n-1}$ is almost surely constant, we need only check if

$$E\left[S_{i,n-1}(\bar{X}_{i,n-1} - \bar{X}_{i,n}) + \sum_{j \in \mathcal{N}'_i} \Delta_{r,j}(n) Z_{j,n} \middle| \mathcal{F}_{n-1}\right] \leq 0.$$

So,

$$
\begin{aligned}
&E\left[S_{i,n-1}(\bar{X}_{i,n-1}-\bar{X}_{i,n})+\sum_{j\in\mathcal{N}_i'}\Delta_{r,j}(n)Z_{j,n}\middle|\mathcal{F}_{n-1}\right]\\
&=E\Bigg[-S_{i,n-1}\left(\sum_{j\in\mathcal{N}_i'}\Delta_{r,j}(n)Z_{j,n}+\Delta_{b,j}(n)(1-Z_{j,n})\right)+\sum_{j\in\mathcal{N}_i'}\Delta_{r,j}(n)Z_{j,n}\Bigg|\mathcal{F}_{n-1}\Bigg]\\
&=E\Bigg[\sum_{j\in\mathcal{N}_i'}\Delta_{r,j}(n)(1-S_{i,n-1})Z_{j,n}-\Delta_{b,j}(n)S_{i,n-1}(1-Z_{j,n})\Bigg|\mathcal{F}_{n-1}\Bigg]
\end{aligned}
$$

Now let

$$
\Delta_{b,j}(n)>\Delta_{r,j}(n)\frac{S_{j,n-1}}{1-S_{j,n-1}}\max_{k\text{ s.t. }j\in\mathcal{N}_k'}\frac{1-S_{k,n-1}}{S_{k,n-1}}
$$

Then, using this and the fact that $E[Z_{j,n}|\mathcal{F}_{n-1}]=S_{j,n-1}$ almost surely, we have

$$
\begin{aligned}
&E\left[S_{i,n-1}(\bar{X}_{i,n-1}-\bar{X}_{i,n})+\sum_{j\in\mathcal{N}_i'}\Delta_{r,j}(n)Z_{j,n}\middle|\mathcal{F}_{n-1}\right]\\
&<E\Bigg[\sum_{j\in\mathcal{N}_i'}\Delta_{r,j}(n)(1-S_{i,n-1})Z_{j,n}+\Delta_{r,j}(n)\frac{S_{j,n-1}}{S_{j,n-1}-1}\\
&\qquad\times\max_{k\text{ s.t. }j\in\mathcal{N}_k'}\frac{1-S_{k,n-1}}{S_{k,n-1}}S_{i,n-1}(1-Z_{j,n})\Bigg|\mathcal{F}_{n-1}\Bigg]\\
&=\sum_{j\in\mathcal{N}_i'}\Delta_{r,j}(n)S_{j,n-1}\left[1-S_{i,n-1}+\frac{S_{i,n-1}}{S_{j,n-1}-1}\max_{k\text{ s.t. }j\in\mathcal{N}_k'}\frac{1-S_{k,n-1}}{S_{k,n-1}}(1-S_{j,n-1})\right]\\
&=\sum_{j\in\mathcal{N}_i'}\Delta_{r,j}(n)S_{j,n-1}\left[1-S_{i,n-1}\left(1+\max_{k\text{ s.t. }j\in\mathcal{N}_k'}\frac{1-S_{k,n-1}}{S_{k,n-1}}\right)\right]\\
&=\sum_{j\in\mathcal{N}_i'}\Delta_{r,j}(n)S_{j,n-1}\left[1-S_{i,n-1}\max_{k\text{ s.t. }j\in\mathcal{N}_k'}\frac{1}{S_{k,n-1}}\right]\\
&=\sum_{j\in\mathcal{N}_i'}\Delta_{r,j}(n)S_{j,n-1}\left[1-\frac{S_{i,n-1}}{\min_{k\text{ s.t. }j\in\mathcal{N}_k'}S_{k,n-1}}\right].
\end{aligned}
$$

Now, note that $j\in\mathcal{N}_i'$, and hence, in particular, $\min_{k\text{ s.t. }j\in\mathcal{N}_k'}S_{k,n-1}\leq S_{i,n-1}$ almost surely. Thus, with our value of $\Delta_{b,j}(n)$ for all $j \in \mathcal{N}_i'$, we have almost surely

$$\begin{aligned} &E\left[S_{i,n-1}(\bar{X}_{i,n-1} - \bar{X}_{i,n}) + \sum_{j \in \mathcal{N}_i'} \Delta_{r,j}(n) Z_{j,n} \middle| \mathcal{F}_{n-1}\right] \\ &< \sum_{j \in \mathcal{N}_i'} \Delta_{r,j}(n) S_{j,n-1} \left[1 - \frac{S_{i,n-1}}{\min_{k \text{ s.t. } j \in \mathcal{N}_k'} S_{k,n-1}}\right] \\ &\leq 0. \end{aligned}$$

Thus, for any $i \in V$, if $\{\Delta_{b,i}(n)\}_{n=1}^{\infty}$ obeys its bound almost surely, the neighbourhood proportion of red balls $\{S_{i,n}\}_{n=1}^{\infty}$ is a strict supermartingale. □

**Corollary 2.2.6. (Network Exposure Supermartingale):** *In a general network $\mathcal{G} = (V, \mathcal{E})$, if the curing policies $\{\Delta_{b,i}(t)\}_{t=1}^{\infty}$ obey the bound*

$$\Delta_{b,i}(n) > \Delta_{r,i}(n) \frac{S_{i,n-1}}{1 - S_{i,n-1}} \max_{k \text{ s.t. } i \in \mathcal{N}_k'} \frac{1 - S_{k,n-1}}{S_{k,n-1}}$$

*almost surely for all nodes $i \in V$, then the network exposure $\{\tilde{S}_n\}_{n=1}^{\infty}$, where $\tilde{S}_n = \frac{1}{N}\sum_{i=1}^{N} S_{i,n}$, is a strict supermartingale with respect to the natural filtration $\{\mathcal{F}_n\}_{n=1}^{\infty}$, i.e.,*

$$E[\tilde{S}_n | \mathcal{F}_{n-1}] < \tilde{S}_{n-1} \qquad \textit{almost surely } \forall n \in \mathbb{Z}_{\geq 1}.$$

It is important to note that the policy for $\{\Delta_{b,i}(t)\}_{t=1}^{\infty}$ used in Theorem 2.2.5 is not a tight lower bound, and hence it is possible that less costly policies exist which will still guarantee that the processes $\{S_{i,n}\}_{n=1}^{\infty}$ are supermartingales. In particular, strategies may exist which obey the fixed budget $\mathcal{B}$ on the amount of curing resources that may be used. However, these results motivate the fact that the search for better policies makes sense, since we know that policies exist that will fight the infection and reduce it on average.

## 2.3 Model Approximations

As previously noted, the dynamics of the network contagion process are complicated, especially when considered on general networks. Even for the classical Polya process stochastic approximations have been widely used; see [42] for a survey. For this reason, in this section we develop two useful approximations to this process on a general network that allow us to shed some light on its asymptotic behaviour.

Throughout this section, unless stated otherwise, we consider general network topologies with $\Delta_{r_i}(t) = \Delta_{b,i}(t) = \Delta$ for all $t \in \mathbb{Z}_{\geq 1}$ and $i \in V$. However, to match the 1-step and $(n,1)$-step distributions, we make the simplifying assumption that the neighbourhood of each node $i$ can be represented as a complete network, i.e., all of its neighbours are connected to one another, in order to apply Lemmas 2.1.1 and 2.1.2.

### 2.3.1 Approximation: Computational Model

We now introduce our first approximation technique, where we approximate the contagion process of each node in the network with a classical Polya urn process.

**Model I. (Computational Model):** *We approximate the dynamics of any node i's contagion process using a classical Polya process* $\mathsf{Polya}(\rho_c = \rho_i, \delta_c = \hat{\delta}_i)$*, with*

$$\rho_i = \frac{\sum_{j \in \mathcal{N}_i'} R_j}{\sum_{j \in \mathcal{N}_i'} T_j}, \quad \text{and} \quad \hat{\delta}_i = \arg\min_{\tilde{\delta}} \frac{1}{n} D\left(P_{i,n}^{(n)} || Q_{\rho_i,\tilde{\delta}}^{(n)}\right),$$

*where*

$$Q_{\rho_i,\tilde{\delta}}^{(n)}(a^n) = \frac{\Gamma\left(\frac{1}{\tilde{\delta}}\right)\Gamma\left(\frac{\rho_i}{\tilde{\delta}} + \bar{a}^n\right)\Gamma\left(\frac{1-\rho_i}{\tilde{\delta}} + n - \bar{a}^n\right)}{\Gamma\left(\frac{1}{\tilde{\delta}} + n\right)\Gamma\left(\frac{\rho_i}{\tilde{\delta}}\right)\Gamma\left(\frac{1-\rho_i}{\tilde{\delta}}\right)}$$

*is the n-fold distribution of the classical Polya process,* $\Gamma(\cdot)$ *is the Gamma function,* $a^n = (a_1, ..., a_n) \in \{0,1\}^n$*, and* $\bar{a}^n = a_1 + \cdots + a_n$*.*

Here $\rho_c$ is chosen to be the proportion of red balls $\rho_i$ in the node's super urn, so that the 1-dimensional distributions of the classical Polya process and the node process $\{Z_{i,n}\}$ coincide as seen in Lemma 2.1.1, while $\hat{\delta}_i$ is set by performing a minimization to find the value that best fits $Q^{(n)}_{\rho_i,\hat{\delta}_i}$ to the distribution of $\{Z_{i,n}\}_{n=1}^{\infty}$ of node $i \in V$. We use a divergence measure, denoted by $D(\cdot||\cdot)$, to observe the quality of the fit.

The explicit derivation of the distribution $Q^{(n)}_{\rho_i,\hat{\delta}_i}$ can be found in [21, 29]. This method ensures that the fit of $Q^{(n)}_{\rho_i,\hat{\delta}_i}$ is as close as possible under the given divergence measure. Since we are measuring the error in using an approximating distribution, we use the Kullback-Leibler divergence [17]; we thus have that

$$\begin{aligned}\hat{\delta}_i &= \arg\min_{\tilde{\delta}} \frac{1}{n} \sum_{a^n \in \{0,1\}^n} P^{(n)}_{i,n}(a^n) \log \frac{P^{(n)}_{i,n}(a^n)}{Q^{(n)}_{\rho_i,\tilde{\delta}}(a^n)} \\ &= \arg\max_{\tilde{\delta}} \frac{1}{n} \sum_{a^n \in \{0,1\}^n} P^{(n)}_{i,n}(a^n) \log Q^{(n)}_{\rho_i,\tilde{\delta}}(a^n)\end{aligned}$$

since $P^{(n)}_{i,n}(a^n) \log P^{(n)}_{i,n}(a^n)$ is independent of $\tilde{\delta}$. The approximating process is stationary and exchangeable, as it is a classical Polya process. We also know (from Section 1.1) that it is non-ergodic with its sample average converging almost surely to the $\mathsf{Beta}(\frac{\rho_i}{\hat{\delta}_i}, \frac{1-\rho)i}{\hat{\delta}_i})$ distribution. Calculating an analytic expression for the minimizing $\hat{\delta}_i$ is not feasible in general, and hence should be performed computationally. However, due to the above minimization, the value of $\hat{\delta}_i$ is, by definition, the best way to fit a Polya process to the process $\{Z_{i,n}\}_{n=1}^{\infty}$ for a given $i$.

### 2.3.2 Approximation: Analytical Models

An alternative to Model I is to attempt to find approximations whose parameters can be determined analytically.

**Model II(a). (Large-Network Analytical Model):** *For any given node $i$, we approximate the dynamics of its process $\{Z_{i,n}\}_{n=1}^{\infty}$ by using a classical Polya process $\mathsf{Polya}(\rho_c = \rho_i, \delta_c = \delta_i')$, with*

$$\rho_i = \frac{\sum_{j \in \mathcal{N}_i'} R_j}{\sum_{j \in \mathcal{N}_i'} T_j}, \quad \text{and} \quad \delta_i' = \frac{\delta_i}{N + (N-1)\delta_i},$$

*where $\delta_i = \frac{N\Delta}{\sum_{j \in \mathcal{N}_i'} T_j}$.*

Here the parameters of the classical Polya process are chosen by directly matching its first and $(n, 1)$-step second-order statistics with those of $\{Z_{i,n}\}_{n=1}^{\infty}$ from Lemmas 2.1.1 and 2.1.2. This method avoids the computational burden of the previous model by yielding an analytical expression for the correlation parameter $\delta_i'$.

We next prove that under some stationarity and symmetry assumptions, the contagion process running on each node in the network is statistically identical to the classical Polya process of Model II(a).

**Lemma 2.3.1. (Exact Representation):** *Suppose that*

*A1.* $P(Z_{i,1} = 1 \mid Z_{j,1}^{n-1} = a^{n-1}) = \rho_i$, *and*

*A2.* $P(Z_{i,t} = 1 | Z_{j,1}^{n-1} = a^{n-1}) = P(Z_{k,n} = 1 | Z_{j,1}^{n-1} = a^{n-1})$,

*for all $n \geq 1, 2 \leq t < n$, $i, j, k \in V$, $a^{n-1} \in \{0,1\}^{n-1}$. Then for any node $i$ in a complete network, $\{Z_{i,n}\}_{n=1}^{\infty}$ is given exactly by the $\mathsf{Polya}(\rho_i, \delta_i')$ process.*

*Proof.* For any node $i$, we wish to show that for all $n$, the $n$-dimensional distributions of $\{Z_{i,n}\}_{n=1}^{\infty}$ and the $\mathsf{Polya}(\rho_i, \delta_i')$ process are identical. It is enough to show that the conditional probability of one event given the whole past is the same, since any joint probability can be written as a product of conditional probabilities. Let us define the events $A_{n-1} = \{Z_{i,1}^{n-1} = a^{n-1}\}$ and $B_{n-1}\left(\{b_{j,1}^{n-1}\}_{j\neq i}\right) = \{Z_{j,1}^{n-1} = b_{j,1}^{n-1}\}_{j\neq i}$. Then,

$$\begin{aligned}
P_{i|n} &:= P(Z_{i,n} = 1 \mid A_{n-1}) \\
&= \sum_{b_{j,1}^{n-1}\in\{0,1\}^{n-1}:j\neq i} P\left(Z_{i,n} = 1 \mid A_{n-1}, B_{n-1}\left(\{b_{j,1}^{n-1}\}_{j\neq i}\right)\right) P\left(B_{n-1}\left(\{b_{j,1}^{n-1}\}_{j\neq i}\right) \mid A_{n-1}\right) \\
&= \sum_{b_{j,1}^{n-1}:j\neq i} \frac{\rho_i + \frac{\delta_i}{N}\sum_{t=1}^{n-1}(a_t + \sum_{j\neq i} b_{j,t})}{1+(n-1)\delta_i} P\left(B_{n-1}\left(\{b_{j,1}^{n-1}\}_{j\neq i}\right) \mid A_{n-1}\right) \\
&= \sum_{b_{j,1}^{n-1}:j\neq i} \frac{\rho_i(1-(N-1)\delta_i') + \delta_i'\sum_{t=1}^{n-1}(a_t + \sum_{j\neq i} b_{j,t})}{1+(N(n-2)+1)\delta_i'} P\left(B_{n-1}\left(\{b_{j,1}^{n-1}\}_{j\neq i}\right) \mid A_{n-1}\right) \\
&= \frac{\rho_i(1-(N-1)\delta_i') + \delta_i'\sum_{t=1}^{n-1} a_t}{1+(N(n-2)+1)\delta_i'} \sum_{b_{j,1}^{n-1}:j\neq i} P\left(B_{n-1}\left(\{b_{j,1}^{n-1}\}_{j\neq i}\right) \mid A_{n-1}\right) \\
&\quad + \frac{\delta_i'}{1+(N(n-2)+1)\delta_i'} \sum_{t=1}^{n-1}\sum_{j\neq i}\sum_{b_{j,1}^{n-1}:j\neq i} b_{j,t} P\left(B_{n-1}\left(\{b_{j,1}^{n-1}\}_{j\neq i}\right) \mid A_{n-1}\right) \\
&= \frac{\left(\rho_i(1-(N-1)\delta_i') + \delta_i'\sum_{t=1}^{n-1} a_t\right)\cdot 1}{1+(N(n-2)+1)\delta_i'} + \frac{\delta_i'\sum_{t=1}^{n-1}\sum_{j\neq i} P(Z_{j,t} = 1 \mid A_{n-1})}{1+(N(n-2)+1)\delta_i'}.
\end{aligned}$$

Then using assumption A1, we have

$$\begin{aligned}
P_{i|n} &= \frac{\rho_i(1-(N-1)\delta_i') + \delta_i'\sum_{t=1}^{n-1} a_t + \delta_i'(N-1)\rho_i + \delta_i'\sum_{t=2}^{n-1}\sum_{j\neq i} P(Z_{j,t} = 1 \mid A_{n-1})}{1+(N(n-2)+1)\delta_i'} \\
&= \frac{\rho_i + \delta_i'\sum_{t=1}^{n-1}\left[a_t + \sum_{j\neq i} P(Z_{j,t} = 1 \mid A_{n-1})\right]}{1+(N(n-2)+1)\delta_i'}
\end{aligned}$$

Now using assumption A2, we get

$$P_{i|n} = \frac{\rho_i + \delta_i'\left(\sum_{t=1}^{n-1} a_t + \sum_{t=2}^{n-1}\sum_{j\neq i} P_{i|n}\right)}{1+(N(n-2)+1)\delta_i'} = \frac{\rho + \delta_i'\left(\sum_{t=1}^{n-1} a_t + (n-2)(N-1)P_{i|n}\right)}{1+(N(n-2)+1)\delta_i'}.$$

Thus, we have that

$$P_{i|n} = \frac{\rho_i + \delta_i'\left(\sum_{t=1}^{n-1} a_t + (n-2)(N-1)P_{i|n}\right)}{1+(N(n-2)+1)\delta_i'} \Rightarrow P_{i|n} = \frac{\rho_i + \delta_i'\sum_{t=1}^{n-1} a_t}{1+(n-1)\delta_i'},$$

which is the conditional probability $P(Z_n = 1|Z_1^{n-1} = a^{n-1})$ for a $\mathsf{Polya}(\rho_i, \delta_i')$ process. A similar calculation can be performed for $P(Z_{i,n} = 0 \mid Z_{i,1}^{n-1} = a^{n-1})$. □

Unfortunately in a general network setting assumptions A1 and A2 above do not hold true. However, this result motivates the fact that this analytical approximation is reasonable to use for situations where these assumptions hold within tolerable margins of error; empirical evidence indicates that this occurs for large values of $N$, since as $N$ increases the quality of the fit improves. This approximation, nevertheless, drastically reduces the complexity in analyzing the individual contagion draw processes, as closed-form expressions for the process parameters are available.

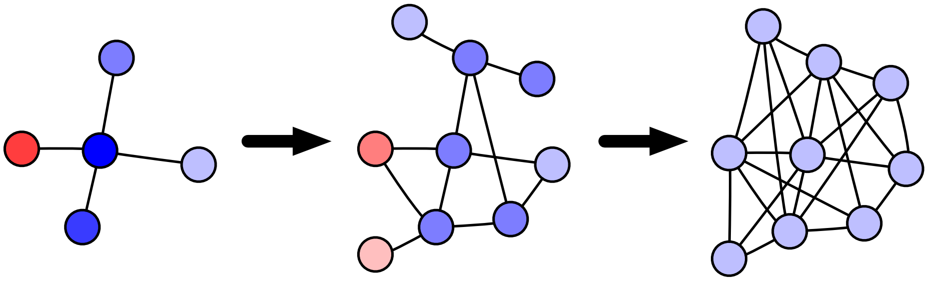


Figure 2.3: Illustration of contagion dilution.

**Observation: (Contagion Dilution)** In situations where the conditions of Lemma 2.3.1 hold within an acceptable range of error, making Model II(a) a good approximation for the nodes' contagion processes, we note that as the number of nodes $N$ increases and the network becomes more connected (in fact, closer to becoming complete), the correlation parameter $\delta'$ decreases. Indeed, $\delta' \to 0$ as $N \to \infty$, and thus the draw variables of the process of each node become independent and identically distributed, since we are simply drawing with replacement. Hence for each node $i$, by the strong law of large numbers, we know that the sample average $\frac{1}{n}\sum_{t=1}^{n} Z_{i,n}$ converges almost surely to a constant, which must be the expected value $E[Z_{i,n}] = \rho$. This means that for complete networks with a large enough number of nodes, the sample average of draws is effectively constant at $\rho$, and so the average infection rate is stable and fixed at $\rho$. This implies that by increasing the number of nodes in the network and by making the network fully connected (see Figure 2.3) so that the conditions of Lemma 2.3.1 hold, we may limit the spread of contagion beyond the initial level of infection $\rho$. The reduction of contagion spread effectively means that all nodes average out their own individual initial infection and share it in the network. For example, a large and highly connected group of healthy nodes and one very infected node will eventually become a group of slightly less healthy individuals, but none will be very infected. This means that the average infection rate $\tilde{I}_n$ will be almost surely less than or equal to $\rho$, but there are no guarantees that it do better, regardless of the initial conditions. One can interpret the outcome of this discussion in the framework of consensus or opinion dynamics, where contagion dilution would drastically reduce the opinion of outliers with extreme views.

**Model II(b). (Small-Network Analytic Model):** *Given any node $i$ in the network with a small to moderate number of nodes, we approximate the dynamics of its contagion process $\{Z_n\}_{n=1}^{\infty}$ using a $\mathsf{Polya}(\rho_i, \delta_i^\star)$ process, where*

$$\rho_i = \frac{\sum_{i=1}^N R_i}{\sum_{i=1}^N T_i}, \quad \text{and} \quad \delta_i^\star = \frac{\delta_i/N}{N + (N-1)\delta_i/N} = \frac{\delta_i}{N^2 + (N-1)\delta_i},$$

*where $\delta_i = \frac{N\Delta}{\sum_{j\in\mathcal{N}_i'} T_j}$.*

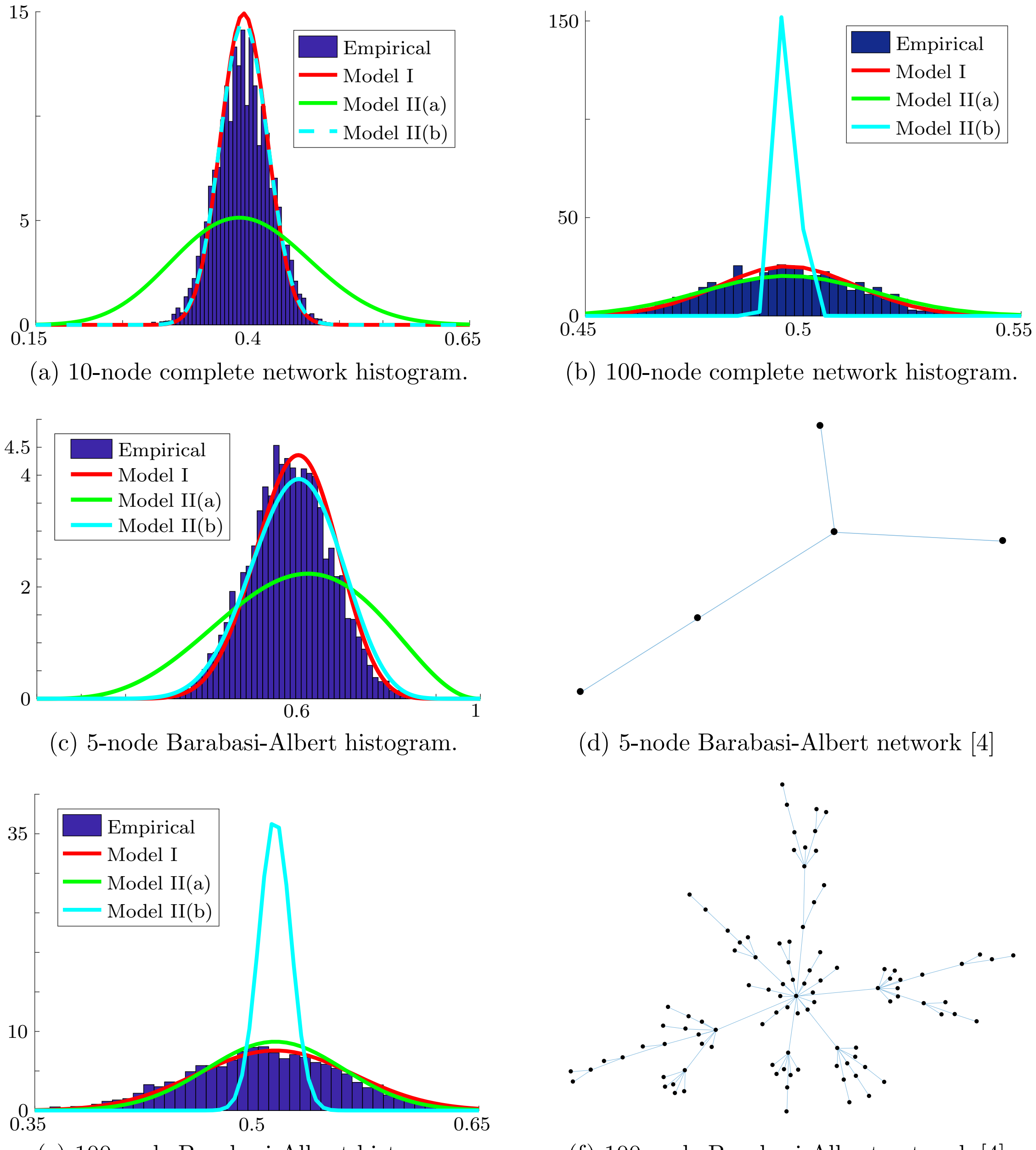
The idea behind this model is that we want to remove the dependence on the number of nodes $N$ from the parameter $\delta_i = \frac{N\Delta}{T_i}$, and so we divide each instance of $\delta_i$ in $\delta_i^\star$ by $N$. Effectively, this means we are using a correlation parameter of $\frac{\Delta}{T_i}$ instead of $\delta = \frac{N\Delta}{T_i}$. The idea is that as $n$ grows, it eventually becomes significantly larger than the relatively small number of nodes $N$, and so $nN \approx n$. Hence, we may consider that for a sufficiently large time, we have added $n\Delta$ balls to the super urn. Simulation results show that this approximation captures the limit distribution of the

Table 2.1: Approximation Usage Scenarios

| Model | Usage Scenario |
|---|---|
| I | Exactness valued over analytic simplicity |
| II(a) | Larger values of $N$, i.e., large network |
| II(b) | Small to moderate values of $N$, i.e., small network |

original process better than Model II(a) when the number of nodes is small. Figure 2.4 displays this relationship. A summary of all models presented in this section, and the scenarios under which they are most suitable, is provided in Table 2.1.

We close this section with numerical demonstrations on the fitness of all models. Figure 2.4 shows a representative comparison between the pdfs of the approximation models and the simulated histogram of $\frac{1}{n}\sum_{t=1}^{n} Z_{i,n}$, where $n = 1000$, for an arbitrary node $i$ in the given networks. Recall that the $\mathsf{Beta}(\frac{\rho_i}{\hat{\delta}_i}, \frac{1-\rho_i}{\hat{\delta}_i})$, $\mathsf{Beta}(\frac{\rho_i}{\delta'_i}, \frac{1-\rho_i}{\delta'_i})$ and $\mathsf{Beta}(\frac{\rho_i}{\delta^\star_i}, \frac{1-\rho_i}{\delta^\star_i})$ pdfs are the distributions of the limit random variables to which the sample average of the draw processes of Models I, II(a) and II(b) (respectively) converge almost surely, as $n \to \infty$ (see Section 1.1). We use complete networks since they satisfy the assumption that all neighbourhoods are complete, as well as Barabasi-Albert networks which have been shown to be a good model for real-world social networks [4] and do not satisfy this assumption; however, our results show that the approximations still fit quite well. As expected, Model I provides the best approximation in all scenarios, albeit without an analytic expression for its parameters which can provide insight into the behaviour of the underlying process. Model II(a) fits quite well when the number of nodes in the network is large, as seen in Figures 2.4b and 2.4e, but fits poorly for a small number of nodes, which is evident in Figures 2.4a and 2.4c. Model II(b) is the complement of Model II(a) in the sense that it fits very well for a small number of nodes but poorly for a large network. Hence if analytic



(a) 10-node complete network histogram.

(b) 100-node complete network histogram.

(c) 5-node Barabasi-Albert histogram.

(d) 5-node Barabasi-Albert network [4]

(e) 100-node Barabasi-Albert histogram.

(f) 100-node Barabasi-Albert network [4]

Figure 2.4: Comparison of normalized simulated histograms for the sample average of draws $\frac{1}{n}\sum_{t=1}^{n} Z_{i,t}$ and $\mathsf{Beta}(\frac{\rho_i}{\hat{\delta}_i}, \frac{1-\rho_i}{\hat{\delta}_i}), \mathsf{Beta}(\frac{\rho_i}{\delta_i'}, \frac{1-\rho_i}{\delta_i'})$, and $\mathsf{Beta}(\frac{\rho_i}{\delta_i^\star}, \frac{1-\rho_i}{\delta_i^\star})$ pdfs from Models I, II(a), and II(b), respectively, for arbitrary nodes with $n = 1000$, averaged over $5,000$ trials. Here the initial conditions $R_i$, $B_i$ and $\Delta_{r,i} = \Delta_{b,i} = \Delta$ were uniformly randomly assigned, but stayed consistent throughout all trials for each network. These simulations took 10 minutes on a 4-core Intel Core i7 processor at 2.20 GHz.

expressions for parameters are desired, Models II(a) and II(b) can be used depending on the number of nodes to provide approximations that are only marginally worse than the computational exactness of Model I.

# Chapter 3

# Control of Epidemics

## 3.1 Problem Statement

The quantities $\{\Delta_{b,i}(n)\}_{n=1}^{\infty}$, which denote the net number of "healthy" balls added to node $i$'s urn after each draw, can play the role of "healing or curing parameters". Our objective is to show that when these parameters are appropriately selected, one can steer the average infection rate towards a desirable level; the selection of curing parameters is, however, subject to an allowable budget on the maximal number of healthy balls that can be added in the network. Let us state this problem formally.

**Problem 3.1.1. (Average Infection Rate Budget Constraint):** *Minimize the limiting average infection rate $\tilde{I}_t$ subject to a budget $\mathcal{B}$ on the total healing at each time step:*

$$\min_{\substack{\sum_{i=1}^{N} \Delta_{b,i}(t) \leq \mathcal{B} \\ \forall t}} \limsup_{t\to\infty} \tilde{I}_t$$

While this is the main problem statement that we will consider in this work, a number of different problems can be considered. A number of candidate problems that were considered are presented below.

**Problem 3.1.2. (Average Infection Rate Threshold):** *Given a desired threshold* $\epsilon \in [0,1]$*, minimize the per-step budget* $\mathcal{B}$ *required to guarantee that the average infection rate* $\tilde{I}_n$ *is below* $\epsilon$*:*

$$\inf \left\{ \mathcal{B} \in \mathbb{R}_{>0} \;\middle|\; \limsup_{t\to\infty} \tilde{I}_t \leq \epsilon \quad \textit{with} \; \sum_{i=1}^{N} \Delta_{b,i}(t) \leq \mathcal{B} \quad \forall t \right\}$$

**Problem 3.1.3. (Fixed Horizon Budget Constraint):** *Minimize the average infection rate* $\tilde{I}_n$ *over a finite time horizon* $[1,T]$ *subject to a budget* $\mathcal{B}$ *on the total healing:*

$$\min_{\sum_{t=1}^{T}\sum_{i=1}^{N}\Delta_{b,i}(t)\leq\mathcal{B}} \tilde{I}_T$$

**Problem 3.1.4. (Fixed Horizon Threshold):** *Given a desired threshold* $\epsilon \in [0,1]$ *and time window* $[1,T]$*, minimize the total budget* $\mathcal{B}$ *required to guarantee that the average infection rate* $\tilde{I}_n$ *is below* $\epsilon$*:*

$$\min \left\{ \mathcal{B} \in \mathbb{R}_{>0} \;\middle|\; \tilde{I}_T \leq \epsilon \quad \textit{with} \; \sum_{t=1}^{T}\sum_{i=1}^{N} \Delta_{b,i}(t) \leq \mathcal{B} \right\}$$

Such optimal curing problems have been studied in many different contexts [40, 35]. For our model, the solution to Problem 3.1.1 would be an infinite horizon optimal control policy that would yield the best possible level of epidemic elimination, given the initial data. Finding such a policy in general appears to be difficult. Nevertheless, as we demonstrate in the upcoming sections, one can obtain interesting analytical results regarding the feasibility of this problem, and design algorithmic strategies to curtail the average infection rate.

The supermartingale results established in the previous section demonstrate the feasibility of a relaxed version of Problem 3.1.1, with no budget limitation. In this section, we establish numerical methods to find control policies that find efficient sub-optimal policies for Problem 3.1.1, under budget constraints and with having

Table 3.1: Curing Strategies

| | |
|---|---|
| (i) | Forcing all $U_{i,n}$ to be supermartingales (Theorem 2.2.3):<br>$\Delta_{bi,}(t) = \frac{\Delta_{r,i}(n)(1-U_{i,n-1})S_{i,n-1}}{U_{i,n-1}(1-S_{i,n-1})}$ |
| (ii) | Forcing all $S_{i,n}$ to be supermartingales (Theorem 2.2.5):<br>$\Delta_{b,i}(t) = \Delta_{r,i}(n)\frac{S_{i,n-1}}{1-S_{i,n-1}} \max_{k \text{ s.t. } i \in \mathcal{N}'_k} \frac{1-S_{k,n-1}}{S_{k,n-1}}$ |
| (iii) | Constrained gradient descent algorithm on a simplex:<br>Find $\Delta_{b,i}(t)$ using Algorithm 2 |
| (iv) | Ratio of degree, closeness centrality and super urn proportion:<br>$\Delta_{b,i}(t) = \mathcal{B}\frac{\lvert\mathcal{N}_i\rvert C_i S_{i,t-1}}{\sum_{j=1}^{N} \lvert\mathcal{N}_j\rvert C_j S_{j,t-1}}$ |
| (v) | Uniformly allocate the budget to all nodes in the network:<br>$\Delta_{b,i}(t) = \frac{\mathcal{B}}{N}$ |
| (vi) | Mixed uniform and $S_{i,n}$ supermartingale strategy:<br>$\Delta_{b,i}(t) = \begin{cases} \Delta_{r,i}(n)\frac{S_{i,n-1}}{1-S_{i,n-1}} \max_{k \text{ s.t. } i \in \mathcal{N}'_k} \frac{1-S_{k,n-1}}{S_{k,n-1}}, & \sum_{s=t-W}^{t} \lvert\Delta_{b,i}(t) - \Delta_r\rvert \geq \epsilon \\ \Delta_{r,i}, & \text{otherwise} \end{cases}$ |

computational complexity in mind. We compare these strategies with the ones obtained from our supermartingale results. A summary of all strategies that will be discussed in this section is given in Table 3.1.

Before we present these strategies in detail, let us describe briefly how we have evaluated their performance. The simulation platform for these strategies is outlined in Algorithm 1. To achieve comparable results, independent trials of the process are run with the same initial conditions $\vec{R} = (R_1, \ldots, R_N)$, $\vec{B} = (B_1, \ldots, B_N)$, and $\vec{\Delta}_r = (\Delta_{r,1}, \ldots, \Delta_{r,N})$ for each curing strategy, and results are then averaged to evaluate expected performance. All results are presented in Chapter 4.

**Algorithm 1** Network contagion curing

```
$A \leftarrow$ adjacency matrix of the network
$\vec{R}, \vec{B}, \vec{\Delta}_r \sim \lceil \mathbf{Uniform}((0, 10]) \rceil^N$
$\mathcal{B} \leftarrow \sum_{i=1}^{N} \Delta_{r,i}$
$\mathit{numCases} \leftarrow$ number of cases, each with a $\mathit{strategy}$
$\mathit{numTrials} \leftarrow$ number of trials to run for each case
$\mathit{steps} \leftarrow$ number of time steps for each trial
for c = 1 : $\mathit{numCases}$ do
    for s = 1 : $\mathit{numTrials}$ do
        $\vec{Z}_{c,s} \leftarrow \textsc{RunTrial}(A, \vec{R}, \vec{B}, \vec{\Delta}_r, \mathcal{B}, \mathit{steps}, \mathit{strategy})$
    $\vec{Z}_c = \frac{1}{\mathit{numTrials}} \sum_{s=1}^{\mathit{numTrials}} \vec{Z}_{c,s}$

procedure $\textsc{RunTrial}(A, \vec{R}, \vec{B}, \vec{\Delta}_r, \mathcal{B}, \mathit{steps}, \mathit{strategy})$
    Initialize $S_{i,0}, U_{i,0}$ using $R_i$ and $B_i$ for all $i \in V$
    for t = 1 : $\mathit{steps}$ do
        Assign $\Delta_{b,i}(t)$ using $\mathit{strategy}$
        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 $S_{i,t}, U_{i,t}$ using $\Delta_{r,i}$ and $\Delta_{b,i}(t)$ for all $i \in V$
```

### 3.2 Supermartingale Strategies

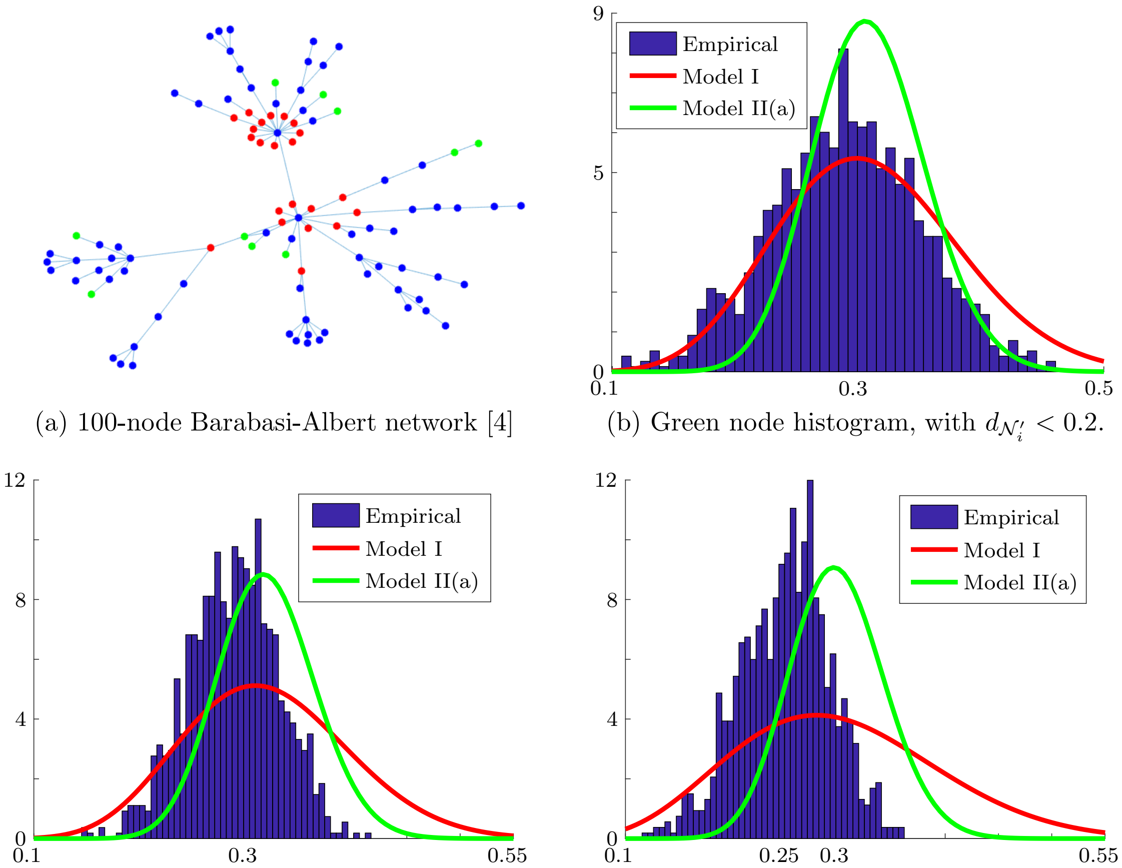
The supermartingales results that we have obtained in Section 2.2, specifically Theorems 2.2.3 and 2.2.5, naturally lead to a class of curing strategies. In particular, these strategies guarantee that $\tilde{U}_n$ and $\tilde{S}_n$, respectively, are supermartingales. It is worth reminding that our theoretical results do not necessarily imply that average infection rate $\tilde{I}_n$ forms a supermartingale. In spite of this, these strategies are still viable options for curing, as far as enough resources are available. We next describe the differences between the strategy given by individual urn proportions, and the one given by super urn proportions.

By Corollary 2.2.4, we know that strategy (i) guarantees that the network susceptibility $\tilde{U}_n$ will be a supermartingale. Hence we set the curing strategy for each node so that it will force its own individual urn proportion of red balls to be a supermartingale. Since draws are taken from the super urns and not the individual urns, the relationship between the reduction of $\tilde{U}_n$ and $\tilde{I}_n$ is not a strong one and our simulations suggests that this strategy does not appear to offer a large reduction in the average infection rate in general. In contrast, the curing strategy given by Corollary 2.2.6, where we choose our curing strategy to force the super urn proportions of red balls to be supermartingales for all nodes, performs reasonably well.

### 3.2.1 Mixed Uniform Strategy

Due to the form of the $S_{i,n}$ supermartingale strategy (ii), the value of $\Delta_{b,i}(t)$ can become close to $\Delta_{r,i}$. When the values are within a threshold $\epsilon$ for some amount of time, a reasonable approximation is to set $\Delta_{b,i}(t) = \Delta_{r,i}$. We thus introduce the mixed uniform strategy; while $\sum_{s=t-W}^{t} |\Delta_{b,i}(t) - \Delta_r| \geq \epsilon$ we simply use strategy (ii), but when the values are closer than $\epsilon$ over a time window $W$ we set $\Delta_{b,i}(t) = \Delta_{r,i}$. When this is the case for all nodes in a neighbourhood, say $\mathcal{N}_i$, we may estimate the limiting behaviour of node $i$ using the approximation models presented in Section 2.3. While we may try to do this on a per-node basis, it will not work for every node. The condition for these models to apply is that $\Delta_{b,i}(t) = \Delta_{r,i}(t) = \Delta$ in the neighbourhood of the node whose process we wish to approximate. Empirical results suggest that as the time grows, the processes settle and hence more nodes satisfy this condition.

The performance of the mixed uniform strategy is presented in Figure 3.1. To make the condition $\Delta_{b,i}(t) = \Delta_{r,i}(t) = \Delta$ easier to satisfy, we set $\Delta_{r,i} = \Delta_r$ for all



(a) 100-node Barabasi-Albert network [4]

(b) Green node histogram, with $d_{\mathcal{N}_i'} < 0.2$.

(c) Blue node histogram, with $0.2 \leq d_{\mathcal{N}_i'} < 1$.

(d) Red node histogram, with $d_{\mathcal{N}_i'} \geq 1$.

Figure 3.1: Comparison of normalized simulated histograms for the mixed uniform strategy and pdfs from Models I and II(a), for arbitrary nodes with $n = 5,000$, averaged over $1,000$ simulated trials. The simulations presented here were set up identically to those in Figure 2.4 and altogether took 20 minutes on a 4-core Intel Core i7 processor at 2.20GHz.

$i \in V$. We set $\epsilon = 0.2$ and $W = 50$, and so we use strategy (ii) until $\sum_{s=t-50}^{t} |\Delta_{b,i}(t) - \Delta_r| < 0.2$, at which time we put $\Delta_{b,i}(t) = \Delta_r$. In Figure 3.1a we show the neighbourhood difference between $\Delta_b$ and $\Delta_r$, $d_{\mathcal{N}_i'} := \sum_{j \in \mathcal{N}_i'} \sum_{s=t-50}^{t} |\Delta_{b,j}(t) - \Delta_r|$. We colour the nodes based on the value of this difference. If $d_{\mathcal{N}_i'} < 0.2$ then node $i$ is coloured green since our condition is met; if $0.2 \leq d_{\mathcal{N}_i'} < 1$ then it is blue and our condition is moderately close; if $d_{\mathcal{N}_i'} > 1$ then it is red since the condition is not met.

While these strategies guarantee a reduction in their respective measures, they use an arbitrary amount of curing resources to do so in general. In fact, as we will see later, these strategies always use a large amount of curing resources relative to the impact they have on reducing the average infection rate. To stay within the framework of Problem 3.1.1, we will now examine a numerical curing strategy that obeys a fixed budget on the per-step curing resources.

## 3.3 Gradient Flow Methods

In this section, we employ the well-known gradient descent algorithm [14] for Problem 3.1.1. As discussed earlier, using $\tilde{I}_n$ as a measure of infection is computationally expensive, and hence we instead focus on the network exposure $\tilde{S}_n$. While our suggested gradient descent algorithm will not provide the exact answer to Problem 3.1.1 for reducing $\tilde{I}_n$, we will show that it is guaranteed to provide the optimal policy to reduce the closely related measure $\tilde{S}_n$.

In Problem 3.1.1, our curing policy is constrained by a budget $\mathcal{B}$ at each time step and so the feasible set, or set of valid curing policies, for our gradient descent is all policies which do not exceed $\mathcal{B}$. However, any optimal policy will make use of the whole budget, and so our feasible set is $\mathcal{X} = \Big\{\{\Delta_{b,i}(n)\}_{i=1}^{N} \in \mathbb{R}_{\geq 0}^{N} \mid \sum_{i=1}^{N} \Delta_{b,i}(n) = \mathcal{B}\Big\}$. Proposition 3.3.1 shows that for arbitrary initial conditions and network topologies, the problem under study for the expected network exposure $E[\tilde{S}_n|\mathcal{F}_{n-1}]$ is convex.

**Proposition 3.3.1. (Gradient descent conditions are met):** *In a general network $\mathcal{G} = (V, \mathcal{E})$ with arbitrary initial conditions, the expected network exposure $E[\tilde{S}_n|\mathcal{F}_{n-1}]$ is convex with respect to the curing parameters $\{\Delta_{b,i}(n)\}_{i=1}^N$ for all $n$. Furthermore, the feasible set*

$$\mathcal{X} = \left\{ \{\Delta_{b,i}(n)\}_{i=1}^N \in \mathbb{R}^N_{\geq 0} \;\middle|\; \sum_{i=1}^N \Delta_{b,i}(n) = \mathcal{B} \right\}$$

*is convex and compact.*

*Proof.* First note that as a function of the parameters $x = (\Delta_{b,1}(n), \ldots, \Delta_{b,N}(n))$, $E[\tilde{S}_n|\mathcal{F}_{n-1}]$ is of the form

$$f_n(x) = \frac{1}{N} \sum_{i=1}^N \frac{c_i}{d_i + \sigma_i(x)}$$

where from (2.3), we can see that

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

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

$$\sigma_i(x) = \sum_{j \in \mathcal{N}_i'} x_j (1 - E[Z_{j,n}|\mathcal{F}_{n-1}]).$$

Note that a number of the variables above are random, but are almost surely constant given $\mathcal{F}_{n-1}$. We thus need to show that, for $x, y \in \mathbb{R}^N_{\geq 0}, \lambda \in [0,1]$,

$$f_n\left(\lambda x + (1-\lambda) y\right) \leq \lambda f_n(x) + (1-\lambda) f_n(y).$$

Firstly, note that

$$\begin{aligned}\sigma_i(\lambda x + (1-\lambda)y) &= \sum_{j \in \mathcal{N}_i'} \left[\lambda x + (1-\lambda) y\right]_j \left(1 - E[Z_{j,n}|\mathcal{F}_{n-1}]\right) \\ &= \sum_{j \in \mathcal{N}_i'} (\lambda x_j + (1-\lambda) y_j)(1 - E[Z_{j,n}|\mathcal{F}_{n-1}])\end{aligned}$$

$$
\begin{aligned}
&= \lambda \sum_{j \in \mathcal{N}_i'} x_j(1 - E[Z_{j,n}|\mathcal{F}_{n-1}]) + (1-\lambda) \sum_{j \in \mathcal{N}_i'} y_j(1 - E[Z_{j,n}|\mathcal{F}_{n-1}]) \\
&= \lambda\sigma_i(x) + (1-\lambda)\sigma_i(y).
\end{aligned}
$$

Then,

$$
\begin{aligned}
&f_n\left(\lambda x + (1-\lambda)y\right) - \lambda f_n(x) - (1-\lambda) f_n(y) \\
&= \frac{1}{N}\sum_{i=1}^{N} \frac{c_i}{d_i + \sigma_i(\lambda x + (1-\lambda)y)} - \frac{\lambda c_i}{d_i + \sigma_i(x)} - \frac{(1-\lambda)c_i}{d_i + \sigma_i(y)} \\
&= \frac{1}{N}\sum_{i=1}^{N} \frac{c_i}{d_i + \lambda\sigma_i(x) + (1-\lambda)\sigma_i(y)} - \frac{\lambda c_i}{d_i + \sigma_i(x)} - \frac{(1-\lambda)c_i}{d_i + \sigma_i(y)} \\
&= \frac{1}{N}\sum_{i=1}^{N} \frac{c_i(d_i + \sigma_i(y))(d_i + \sigma_i(x))}{(d_i + \lambda\sigma_i(x) + (1-\lambda)\sigma_i(y))(d_i + \sigma_i(x))(d_i + \sigma_i(y))} \\
&\qquad - \frac{c_i(\lambda(\sigma_i(y) - \sigma_i(x)) + d_i + \sigma_i(x))(d_i + \lambda\sigma_i(x) + (1-\lambda)\sigma_i(y))}{(d_i + \lambda\sigma_i(x) + (1-\lambda)\sigma_i(y))(d_i + \sigma_i(x))(d_i + \sigma_i(y))} \\
&= \frac{1}{N}\sum_{i=1}^{N} \frac{c_i\lambda(\lambda - 1)(\sigma_i(x) - \sigma_i(y))^2}{(d_i + \lambda\sigma_i(x) + (1-\lambda)\sigma_i(y))(d_i + \sigma_i(x))(d_i + \sigma_i(y))} \\
&\leq 0,
\end{aligned}
$$

since $\lambda - 1 \leq 0$ and all other terms are nonnegative. Hence $E[\tilde{S}_n|\mathcal{F}_{n-1}]$ is convex in the curing parameters $(\Delta_{b,1}(n), \ldots, \Delta_{b,N}(n))$ for all time. Lastly, the constraint set $\left\{\{\Delta_{b,i}(n)\}_{i=1}^{N} \in \mathbb{R}_{\geq 0}^{N} \mid \sum_{i=1}^{N} \Delta_{b,i}(n) = \mathcal{B}\right\}$ is clearly a finite-dimensional simplex and hence convex and compact. □

The structure of the feasible set $\mathcal{X}$ allows us to employ the simplex constrained gradient descent method, see [14, Chapter 2]; this procedure is fully described in Algorithm 2. The time complexity of this algorithm is of the order $O(sa)$ at each time step, where $s$ is the stopping time of the gradient descent and $\frac{1}{a}$ is the granularity used to find the limit-minimized step size $\alpha_k$. While Proposition 3.3.1 guarantees that

**Algorithm 2** Constrained gradient descent on a simplex [14]

```
Start at an arbitrary node:
$y_1 = (\mathcal{B}, 0, \ldots, 0)$

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

    Move only in that direction:
    $[\bar{y}_k]_i = \mathcal{B}$, 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:
    $y_{k+1} = y_k + \alpha_k(\bar{y}_k - y_k)$
```

the curing policy that this algorithm finds will be optimal for each individual step, it does not guarantee optimality over the entire time horizon. In spite of this, as the simulation results in Figure 4.1 show, this curing strategy still outperforms all other curing strategies studied in this paper. The downside of the gradient method is that it is computationally expensive to execute, as it requires intimate knowledge of the state of all nodes in the network. This motivates us to seek other methods which are computationally easier to execute, although they do not perform as well as the gradient descent strategy.

## 3.4 Heuristic Strategies

Both sets of strategies identified above come with challenges. The supermartingale strategies are accompanied by analytical results that guarantee that they will improve in general, but they do not obey a fixed budget, nor do they create a significant reduction in the average infection rate. The gradient flow method uses a fixed budget and is provably optimal to reduce the expected network exposure $E[\tilde{S}_n|\mathcal{F}_{n-1}]$, but it is computationally costly and requires a large amount of information about the state of infection at every node, including the entire history of draws and values of the curing parameters. As a compromise between these strategies we present the centrality-infection ratio strategy, which is a heuristic centrality-based strategy designed to allocate the fixed per-step budget $\mathcal{B}$.

The idea is to create a ratio to split the budget between all nodes in the network, whose time complexity will be of the order $O(1)$. We consider three factors when determining how much curing a node should receive: local impact, topological position, and level of infection. Nodes with higher local impact have more neighbours, and hence any healing they receive is immediately distributed to a larger number of nodes. Those with a better topological position are more centrally located within the network, in the sense that the distance from them to all other nodes is smaller. Lastly, nodes with a higher level of infection will need more curing resources to become healthy.

The resulting curing strategy, which we call the centrality-infection ratio, is

$$\Delta_{b,i}(t) = \mathcal{B}\frac{|\mathcal{N}_i|C_iS_{i,t-1}}{\sum_{j=1}^{N}|\mathcal{N}_j|C_jS_{j,t-1}}.$$

To measure local impact of node $i$, we use the degree, $|\mathcal{N}_i|$, which measures the number of neighbours for node $i$. Topological position is determined by calculating the closeness centrality [12], which, for node $i$, is defined as

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

where $d(i,j)$ is the length of the shortest path from node $i$ to node $j$. Thus $C_i$ will be higher than $C_j$ if node $i$ is closer to all other nodes than node $j$, in the sense that the paths from node $i$ will be shorter in total. Hence $C_i$ gives more importance to the nodes which are more central, and thus have more influence on the overall network. Finally, to measure the level of infection, we use the super urn proportion of red balls $S_{i,n}$. From (2.4), we know that this quantity captures how likely it is for node $i$ to be infected at this time given the history of the process. Thus we give more importance to nodes who are more likely to be infected, so that we may make them less likely to be infected in the future.

The advantage of this heuristic strategy is twofold. Not only does it reduce computational time complexity from $O(sa)$ to $O(1)$, it is also somewhat distributed in the sense that it does not require constant information from the entire network. Unlike the gradient descent algorithm, strategy (iv) simply needs to know information about the network topology and the state of infection of each node. Since we assume that our network's graph is constant in time, this topological information is only required initially and can be used thereafter. The only other information required from the network at large is the sum of the super urn ratios $\sum_{i=1}^{N} S_{i,n}$, and hence much less information needs to be communicated through the network for the implementation of this strategy.

Lastly, for comparison purposes we present the uniform curing strategy (v), which splits the budget $\mathcal{B}$ equally to all nodes in the network. We use this strategy as a benchmark to show the improvement achieved by intelligently assigning the curing resources.

# Chapter 4

# Simulation Results

## 4.1 Simulation Setup

In order to confirm the results of Theorems 2.2.3 and 2.2.5, a number of simulations were performed; the pseudocode is outlined in Algorithm 1. While the simulations performed had the numbers of red balls added $\Delta_{r,i}$ vary between nodes, they were constant in time. This was done to simplify the choice of the per-step budget, and does not affect the execution of the simulations themselves. All initial conditions for the simulations herein, as well as videos displaying the average performance of the curing strategies, are available online.[1]

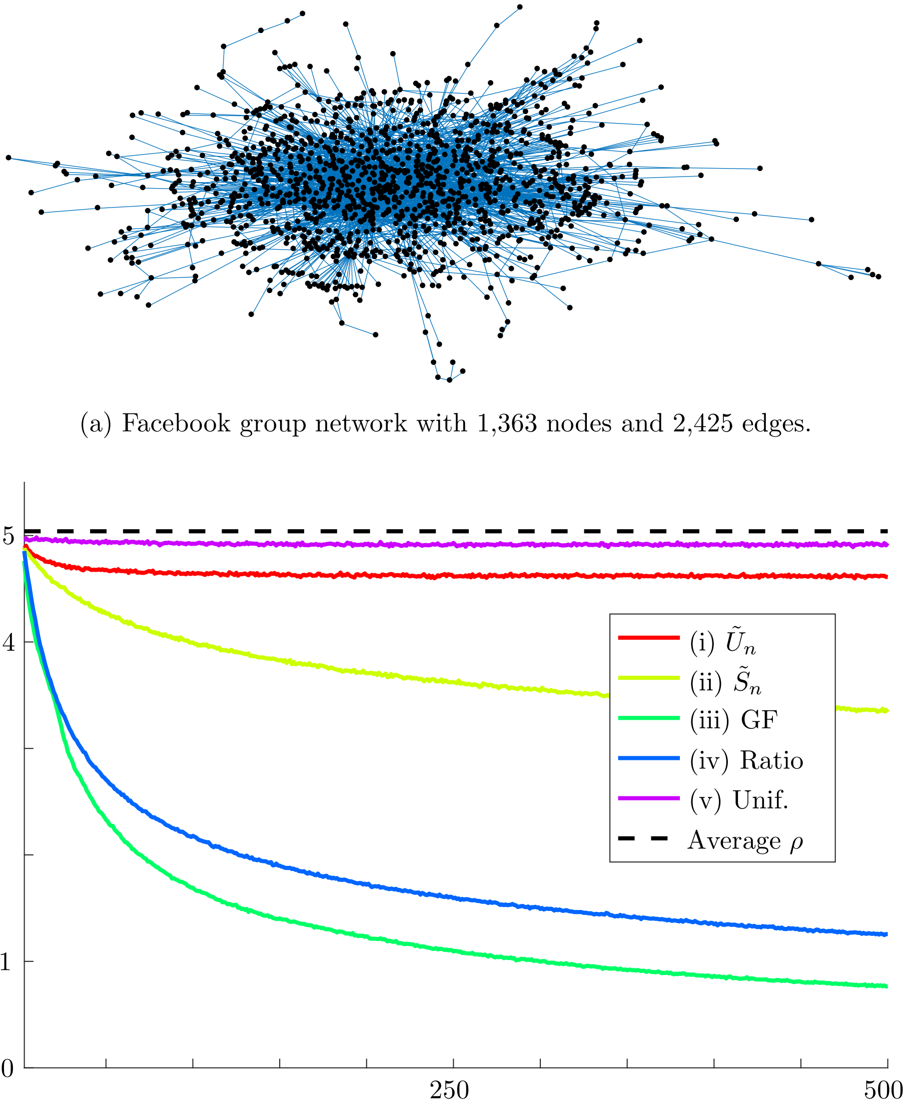
The network shown in Figure 4.1a was generated by using a tool [43] to crawl through 500 posts in a Facebook group. Individuals who created posts or interacted with others' content are represented by nodes, while edges are created if individuals interacted with the post or comment of another (by commenting on the post, or liking the post or comment). The resulting graph has 1,363 nodes and 2,425 edges, and by design represents the topology of a real social network.

[1] See: http://bit.ly/2szl8PY

We now provide a detailed description of the simulation, as described in Algorithm 1. The values of $R_i$, $B_i$ and $\Delta_{r,i}$ were uniformly randomly assigned for each node as integers between 1 and 10. These values remained consistent for all strategies and throughout all trials that were performed. Since the values for $\Delta_{r,i}$ were fixed over time, the per-step budget was set at $\mathcal{B} = \sum_{i=1}^{N} \Delta_{r,i}$. With the initial conditions set, a number of trials were performed for each strategy. Each trial was performed by successively drawing balls from super urns for a fixed number of time steps. At time $t$, we first assigned the curing $\Delta_{b,i}(t)$ based on the strategy selected. Then a uniform random variable on $[0, 1]$, $Y_i$, was generated for each node $i$ and compared to the super urn proportion. If $Y_i < S_{i,t-1}$ then we say that a red ball was drawn and so $Z_{i,t} = 1$, otherwise we drew black and so $Z_{i,t} = 0$. Based on what was drawn, we added $\Delta_{r,i}$ red or $\Delta_{b,i}(t)$ black balls into node $i$'s urn, and hence its super urn and those of its neighbours. At the end of each trial the draw variables were saved, and then averaged over all trials to produce the empirical performance of the strategy.

## 4.2 Discussion of Simulation Results

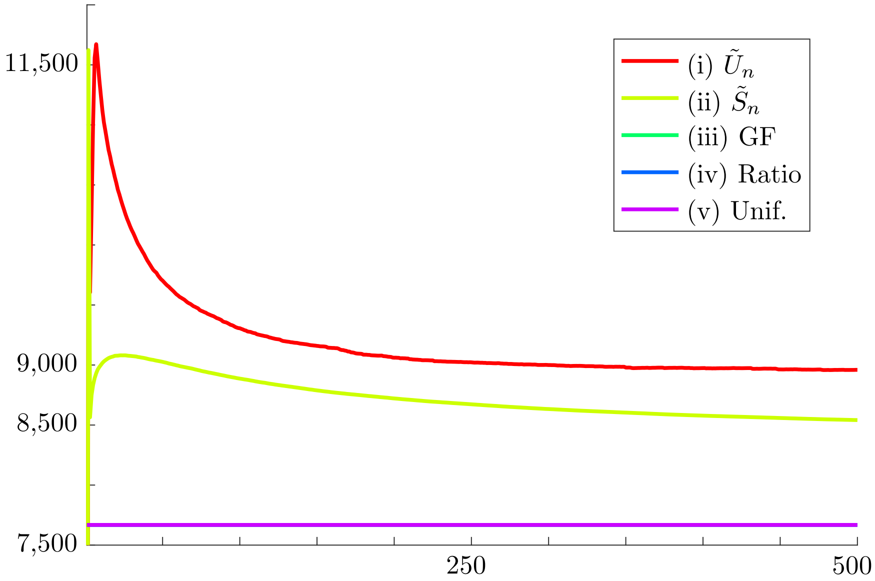
In this section we present results obtained by running simulations using Algorithm 1. Figure 4.1 displays results on the Facebook network with all curing strategies, except the mixed uniform, for a short time window due to computational limitations with the gradient descent strategy (iii). Figure 4.2 shows all other strategies for a longer time frame, although the conclusions remain the same. Figure 4.3 shows the initial and final individual states of a Barabasi-Albert network using the centrality-infection ratio and uniform curing methods. Finally, in Figure 4.4, we see observe curing performance on the Facebook network when the process has finite memory.



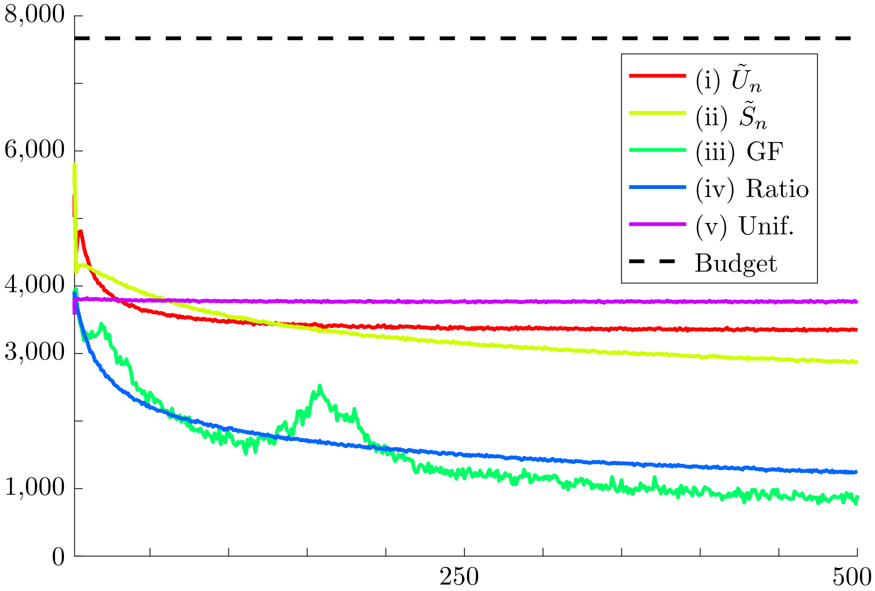
(a) Facebook group network with 1,363 nodes and 2,425 edges.

(b) Plot of empirical average infection rate $\tilde{I}_n$ compared to the network-wide initial level of infection $\rho$ as defined in Equation (2.1) (lower means less infection).

Figure 4.1: Comparison of curing strategies in Table 3.1 except mixed uniform. Simulation results were averaged over 250 trials for each strategy, and altogether took approximately 49 hours on 10 cores of an Intel Xeon processor at 2.20GHz. Initial numbers of balls $R_i$ and $B_i$, and numbers of red balls added $\Delta_{r,i}$ (which remained constant in time), were uniformly randomly assigned for each node but stayed consistent throughout all trials and strategies, while the assignments of $\{\Delta_{b,i}(t)\}_{t=1}^{\infty}$ were different for each strategy. Since the $\Delta_{r,i}$ are constant, the budget was set as $\mathcal{B} = \sum_{i=1}^{N} \Delta_{r,i}$.



(c) Plot of empirical average usage of curing resources, $\sum_{i=1}^{N} \Delta_{b,i}(t)$ (lower means less curing resources used). Note that strategies (iii), (iv) and (v)'s usages are fixed at the per-step budget $\mathcal{B}$, and are overlaid.



(d) Plot of empirical average wasted curing, $\sum_{i=1}^{N} \sum_{t=1}^{n} \Delta_{b,i}(t) Z_{i,t}$ (lower means less curing resources assigned to nodes that did not use them).

Figure 4.1: (continued) Comparison of curing strategeies.

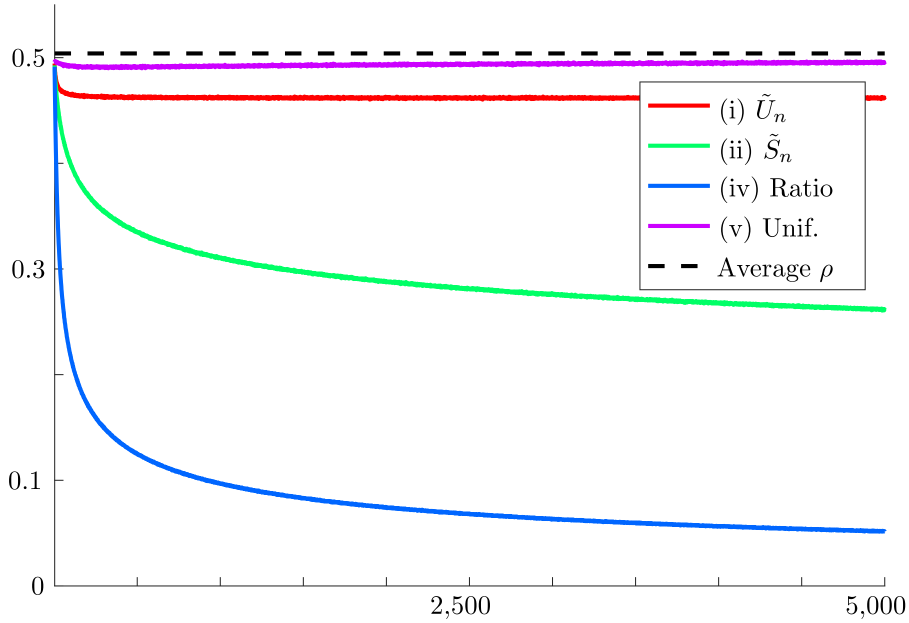


Figure 4.2: Plot of empirical average infection rate $\tilde{I}_n$ on the network shown in 4.1a for a longer time frame. Strategies used are shown in Table I; (iii) was omitted due to computational constraints. The simulations presented here were performed identically to those described in Figure 4.1, with all initial conditions consistent between trials and strategies. Results were averaged over 1,000 trials, and altogether took 30 hours on 12 cores of an Intel Xeon processor at 2.20GHz.

Figures 4.1 and 4.2 show comparisons between the strategies outlined in Table 3.1. It is important to note that only strategies (iii), (iv) and (v) in Table I have a budget $\mathcal{B}$ on the amount of curing they can use. The other two strategies are allowed to vary the total curing they use in time; the amount of resources each strategy consumes is shown in Figure 4.1c. Figure 4.1d displays the average wasted curing resources for each strategy.

Figures 4.1b and 4.2 compare the performance of all strategies described in Table 3.1 on a Facebook network. Figure 4.1b includes the gradient flow algorithm, while Figure 4.2 shows all other strategies over a longer time horizon. The benchmark uniform strategy (v) performs the worst, which is to be expected. Although

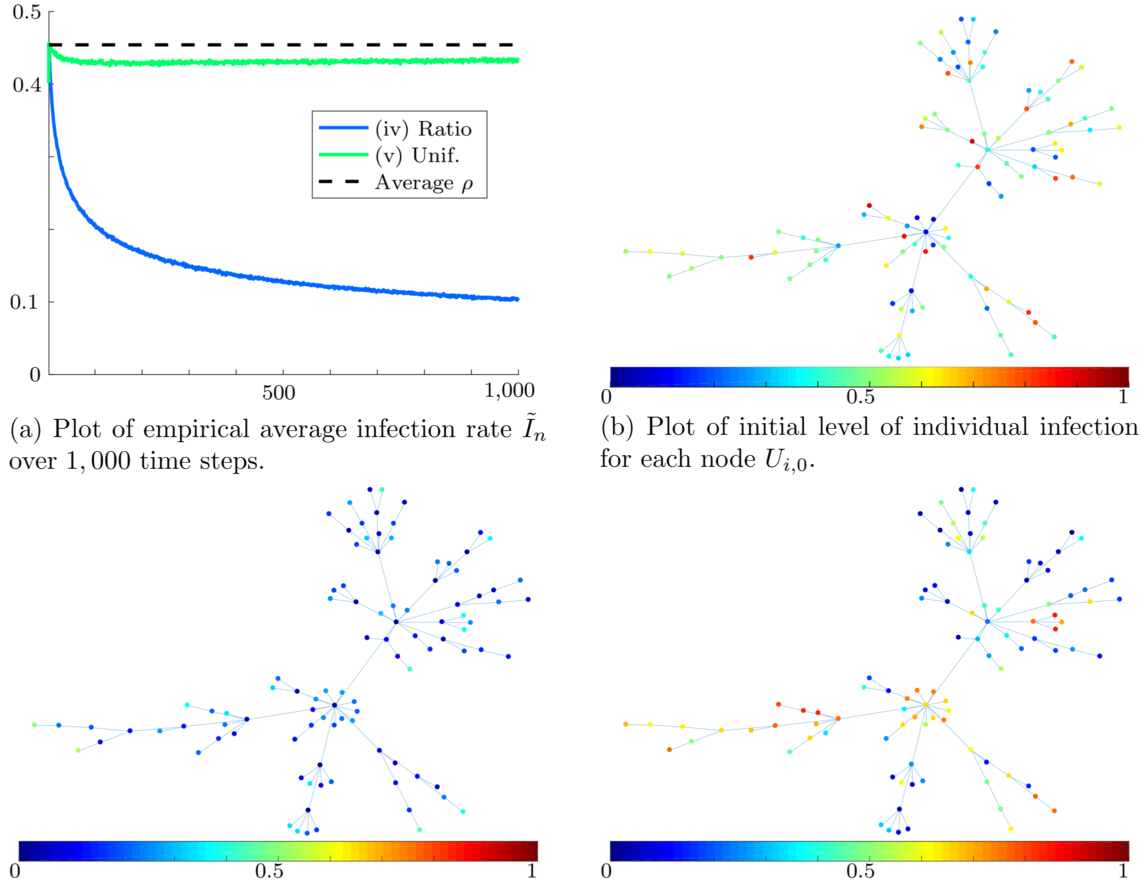
(iii) is only proven to be optimal for the expected network exposure $E[\tilde{S}_n|\mathcal{F}_{n-1}]$, these results are seen to be effective for the average infection rate $\tilde{I}_n$ as well; as we mentioned before, this strategy outperforms all other curing strategies described in this paper. However, the heuristic strategy (iv) performs similarly while being far less computationally difficult. The supermartingale strategies (i) and (ii) both reduce $\tilde{I}_n$ below the initial average infection rate in the network $\rho$, but are less effective in doing so than the gradient flow and centrality-infection ratio methods. Strategy (i) sees only an immediate small reduction in $\tilde{I}_n$, while strategy (ii) continually decreases $\tilde{I}_n$. Hence it appears that forcing $\tilde{U}_n$ to be a supermartingale is not enough to guarantee a large reduction in the average infection rate $\tilde{I}_n$, while guaranteeing that $\tilde{S}_n$ is a supermartingale does seem to cause a reasonable decrease in $\tilde{I}_n$.

In Figure 4.1c we observe the amount of curing resources used by each strategy. Since strategies (iii), (iv) and (v) all obey a per-step budget constraint their usages are fixed. Both supermartingale strategies, which may use arbitrary amounts of curing resources, initially use a larger amount of curing resources and then reduce their usage. Strategy (i) appears to reduce curing consumption at first but then steadily increase, while strategy (ii) continues to decrease its usage in time. Further, strategy (i) uses almost 50% more curing resources than the budget $\mathcal{B}$ initially, while strategy (ii)'s initial usage is only around 18% higher than $\mathcal{B}$.

The amount of curing resources wasted by each strategy is displayed in Figure 4.1d. Waste is defined as curing resources which were assigned to nodes that did not use them since they displayed "infected" behaviour at that time, and so it is measured as $\sum_{i=1}^{N}\sum_{t=1}^{n}\Delta_{b,i}(t)Z_{i,t}$. There is a correlation between the amount of resources wasted and curing performance; strategies which waste less resources tend to be more effective at reducing the average infection rate $\tilde{I}_n$. However, this does not tell the full story. The gradient flow algorithm has several spikes where it wastes more resources than the centrality-infection ratio (iv), but this does not appear to affect its curing performance. Furthermore, strategy (i) initially wastes less than strategy (ii) even though it uses more curing resources, and it still performs worse with respect to reduction in $\tilde{I}_n$. This suggests that optimal curing strategies not only waste less, but also intelligently allocate their curing resources to make the best use of them.

Figure 4.3 shows the initial and final state of all nodes in a randomly generated network for two different curing strategies. Barabasi-Albert networks are randomly generated through preferential attachment and are widely used in the literature since they have been shown to exhibit the properties of real social networks [4]. In Figure 4.3a we see that for such a network the centrality-infection ratio (iv) dramatically outperforms uniform curing (v), as was the case for the social network shown in Figure 4.1a. After 1,000 time steps, strategy (iv) reduced the average infection rate $\tilde{I}_n$ to about 15%, and no node had an individual level of infection above 55%. In comparison, strategy (v) barely reduced $\tilde{I}_n$ below the initial average infection $\rho$, and the individual infection of some nodes was above 90%. This result illustrates the fact that intelligent allocation of curing resources is not only important to reduce the network-wide average infection rate, but the infection of individual nodes as well.

Lastly, in Figure 4.4 we examine the process with finite memory, where added balls remain for only 50 time steps. The comparison of curing performance show in Figure 4.4a is similar to the infinite memory case; however, the average infection rate $\tilde{I}_n$ eventually stays constant in time for all strategies except the gradient flow (iii), which sees a small increase. This suggests that the network Polya contagion process



(a) Plot of empirical average infection rate $\tilde{I}_n$ over $1,000$ time steps.

(b) Plot of initial level of individual infection for each node $U_{i,0}$.

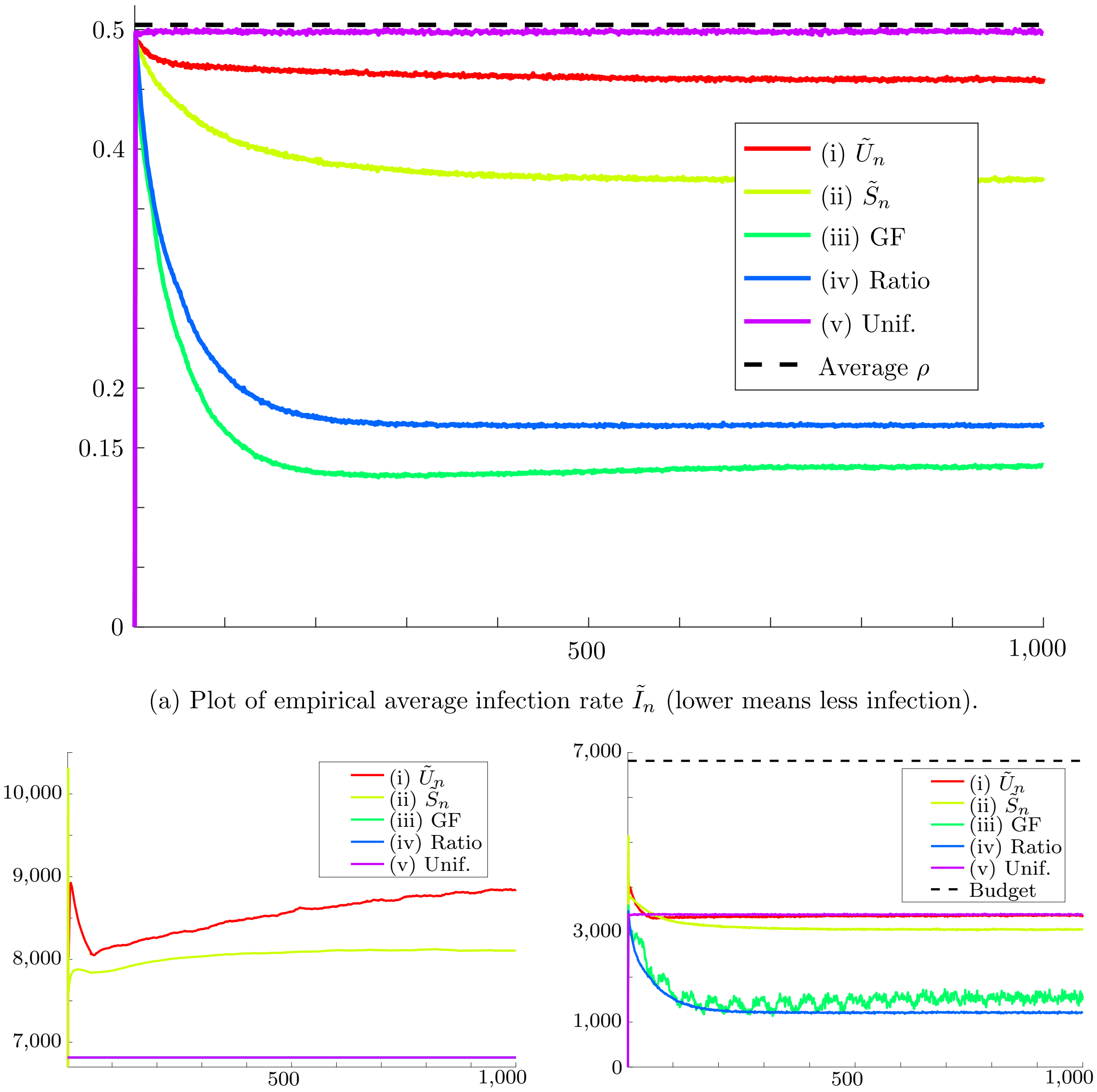
(c) Plot of final individual infection for each node $U_{i,n}$ using centrality-infection ratio (iv).

(d) Plot of final individual infection for each node $U_{i,n}$ using uniform curing (v).

Figure 4.3: Comparison of curing strategies (iv) and (v) on a Barabasi-Albert network [4] with 100 nodes and 99 edges. Here blue represents total healthiness ($U_{i,n} = 0$) while red represents total infection ($U_{i,n} = 1$). Results were averaged over 1,000 trials for each strategy, and altogether took 5 minutes on a 4-core Intel Core i7 processor at 2.20 GHz. This simulation was performed identically to those in Figure 4.1, with all initial conditions consistent between trials and strategies.

with finite memory may be stationary, and perhaps the initial transition probabilities of the process are not that of its stationary distribution.

The results in Figure 4.4b are quite different than what was seen for the infinite memory process in Figure 4.1c. While both supermartingale strategies initially reduce their usage as seen previously, they then slowly increase their usage in time.



(a) Plot of empirical average infection rate $\tilde{I}_n$ (lower means less infection).

(b) Plot of empirical usage of curing resources, $\sum_{i=1}^{N} \Delta_{b,i}(t)$ (lower means less used). Strategies (iii)–(v)'s usages are overlaid at budget $\mathcal{B}$.

(c) Plot of empirical average wasted curing, $\sum_{i=1}^{N} \sum_{t=1}^{n} \Delta_{b,i}(t) Z_{i,t}$ (lower means less assigned to nodes that did not use them).

Figure 4.4: Plot of empirical average infection rate $\tilde{I}_n$ on the network shown in 4.1a with finite memory of 50 steps. Strategies used are shown in Table I, and simulations were performed identically to those described in Figure 4.1 with all initial conditions consistent between trials and strategies. Results were averaged over 1,000 trials, and altogether took 83 hours on 12 cores of an Intel Xeon processor at 2.20GHz.

Strategy (i) in particular appears to need a steady increase in curing resources to be able to guarantee that the susceptibility $\tilde{U}_n$ remains a supermartingale. In contrast, Figure 4.4c is similar to what was seen for the infinite memory process. Only the gradient flow (iii) exhibits different usage; it wastes more than the centrality-infection ratio (iv), and exhibits many more spikes in waste than in the infinite memory case.

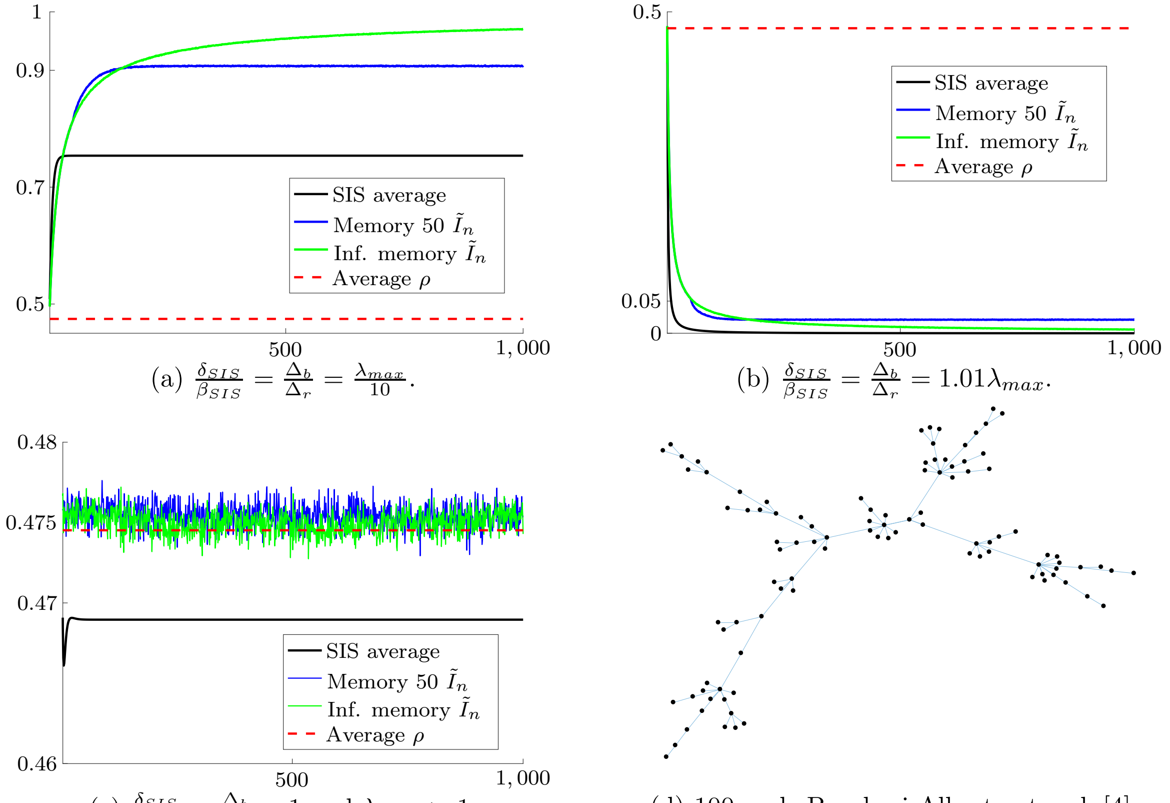
### 4.2.1 Comparison with SIS model

We now provide a number of empirical results through which to compare our model, with both finite and infinite memory, to the traditional discrete time SIS model [46]. In the SIS model, the parameter $\delta_{SIS}$ denotes the probability that a node will recover from infection, and $\beta_{SIS}$ is the probability that a node will become infected through contact with a single infected neighbour. The dynamics are described through the probability that any node $i$ will be infected at time $t$, $P_i(t)$, which evolves according to the equation

$$P_i(t+1) = P_i(t)(1-\delta_{SIS}) + (1-P_i(t))\Big(1 - \prod_{j\in\mathcal{N}_i}(1-\beta_{SIS}P_j(t))\Big).$$

Note in particular that this model exhibits Markovian behaviour, since the evolution of the process depends only on the probability of infection from the previous time step. We make the simplifying assumption that $\delta_{SIS}$ and $\beta_{SIS}$ remain the same for all nodes and throughout time, and hence we will compare it with the network Polya contagion process when $\Delta_r$ and $\Delta_b$ are similarly fixed in time and throughout the network.

The concept of an epidemic threshold for the SIS model gives a value through which one may determine whether the epidemic dies, a priori using only the system parameters [46]. The threshold condition is directly related to the largest-magnitude



(a) $\frac{\delta_{SIS}}{\beta_{SIS}} = \frac{\Delta_b}{\Delta_r} = \frac{\lambda_{max}}{10}$.

(b) $\frac{\delta_{SIS}}{\beta_{SIS}} = \frac{\Delta_b}{\Delta_r} = 1.01\lambda_{max}$.

(c) $\frac{\delta_{SIS}}{\beta_{SIS}} = \frac{\Delta_b}{\Delta_r} = 1$ and $\lambda_{max} > 1$.

(d) 100-node Barabasi-Albert network [4].

Figure 4.5: Comparison between the discrete time SIS model average infection rate $\frac{1}{N}\sum_{i=1}^{N} P_i(n)$ and the network Polya contagion process average infection rate $\tilde{I}_n$. Simulation results were averaged over 5,000 trials, and parameters were randomly assigned but consistent throughout all trials for a given case. Here $\lambda_{max} \approx 5.05$, $\beta_{SIS} = 0.15$ and $\Delta_r = 2$ for all cases, while $\delta_{SIS}$ and $\Delta_b$ were set according to the ratios given above.

eigenvalue $\lambda_{max}$ of the adjacency matrix of the underlying graph of the network, and states that if $\delta_{SIS} > \beta_{SIS}\lambda_{max}$ then the epidemic will be eliminated after some time $n$, i.e., eventually $P_i(t) = 0$ for all $i$ and all $t > n$. Furthermore, it has been shown that this threshold is tight, and indeed if $\delta_{SIS} < \beta_{SIS}\lambda_{max}$ then some non-zero convergence point exists, called an *endemic state*, and the epidemic will never be eliminated [2].

Figure 4.5 compares the behaviour of the SIS model and the network Polya contagion process for different selections of these parameters. The initial probabilities of infection $P_i(0)$ for the SIS model were set as the initial individual proportions of red balls for the nodes $\frac{R_i}{T_i}$. Further, we relate in Figures 4.5a–4.5c the parameters $\beta_{SIS}$ and $\delta_{SIS}$ to $\Delta_r$ and $\Delta_b$, respectively, using ratios of the largest-magnitude eigenvalue $\lambda_{max}$ of the adjacency matrix of the graph shown in Figure 4.5d.

Figure 4.5a shows a comparison when the SIS model is displaying endemic behaviour. We see here that after a very short time the SIS model settles and shortly thereafter the finite memory process settles (albeit to a different value), while for the infinite memory process the individual rates of infection and hence the average $\tilde{I}_n$ continue to increase in time. Since both the SIS model and the finite memory process have limited reinforcement while the infinite memory process does not, these results are to be expected. Figure 4.5b displays a comparison where the epidemic threshold is met and the epidemic dies for the SIS model. Here we see that $\tilde{I}_n$ for both the infinite and finite memory processes decreases and approaches zero, albeit not as quickly as the SIS model. Hence we see that when the curing parameter $\Delta_b$ is much larger (in fact, more than five folds larger) than the infection parameter $\Delta_r$ the epidemic is eliminated, as we expect, and this behaviour of the SIS model is captured by the network Polya contagion process. However, the finite memory process does not fully approach zero, since the initial conditions $R_i$ and $B_i$ have a larger affect due to the lower number of balls in all urns relative to the infinite memory process. Finally, Figure 4.5c shows the case where the epidemic does not die and the parameters in both models are set to be equal. We observe a similar trend between all models, with the finite and infinite memory processes exhibiting near-identical behaviour.

Through these observations, we may conclude that both versions of the network Polya contagion process may apply to the modelling of epidemics, albeit in different applications. The finite memory process exhibits behaviour that is more closely related to the SIS model since they are both limited reinforcement processes, and hence it may be best suited to traditional biological diseases. The infinite memory process obeys similar trends, but in the endemic state there are some interesting differences since the effects of the infection continue to spread throughout the population as they continue to interact with one another, whereas the SIS model quickly settles and does not change in time. Thus with infinite memory the network Polya contagion process is better suited to modelling opinion dynamics, the spread of ideas, and advertising schemes.

# Chapter 5

# Conclusion

In this work we introduced and described the network Polya contagion model, which can be used for the analysis and control of epidemics on networks. We described the process and its stochastic properties, giving expressions for the network-wide probability of infection spread as well as a number of marginal distributions. We showed that it is not stationary and hence not exchangeable in general, but displays some characteristics of asymptotic stationarity. We showed that with finite memory the entire process is Markovian, while the individual node processes are quasi-Markovian in the entire process. We presented a number of approximation models to estimate the limiting behaviour of the process, and displayed their fitness through simulation results.

We formulated an optimal control problem and provided analytical results that show when a relaxed problem has a limit. We presented and compared theoretical, numerical, and heuristic curing strategies to control the epidemic. Lastly, through extensive simulation results we displayed the performance and behaviour of all curing strategies.

A number of different future problems can be considered with this model. Firstly, curing problems that focus on a finite time window or a budget over a fixed horizon could be explored. The allocation of a large fixed budget over a time horizon is more realistic in the context of a viral infection, where a fixed supply of medicine may be available. The effect of network topology on the process could be further explored; for example, bipartite and star graphs could have interesting results. The level of symmetry in bipartite graphs is similar to the complete graph, and so notions such as exchangeability in the nodes instead of in time could be explored. The process with finite memory could be studied more closely, for example to determine whether it can be made stationary. The case of $\Delta_{r,i}(t)$ varying in time could be explored in further detail. Furthermore, the replacement matrix $M_{R,i}(t)$ could be modified to allow nonzero terms in the off-diagonal entries. This could even be examined from a game-theoretic perspective, wherein two decision makers attempt to shift the state of the network to one extreme: either red or black. Finally, a consensus problem could be formulated, with opinions being shared by the network Polya contagion process.

# 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] H. J. Ahn and B. Hassibi. Global dynamics of epidemic spread over complex networks. In *2013 Conf. on Decision and Cont.*, pages 4579–4585, 2013.

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

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

[5] G. Aletti and A. Ghiglietti. Interacting generalized Friedmanś urn systems. *Stochastic Processes and their Applications*, 127:2650–2678, 2017.

[6] G. Aletti, C. May, and P. Secchi. A central limit theorem, and related results, for a two-color randomly reinforced urn. *Adv. Appl. Prob.*, 41:829–844, 2009.

[7] D. Ait Aoudia and F. Perron. A new randomized Polya urn model. *Appl. Mathematics*, 3:2118–2122, 2012.

[8] R. Ash and C. Doléans-Dade. *Probability and Measure Theory.* Academic Press, 2000.

[9] K. B. Athreya and S. Karlin. Embedding of urn schemes into continuous time markov branching processes and related limit theorems. *Ann. Math. Stat.*, 39(6):1801–1817, 1968.

[10] S. Balaji and H. M. Mahmoud. Exact and limiting distributions in diagonal Polya processes. *Annals of Inst. Stat. Math.*, 58:171–185, 2006.

[11] A. Banerjee, P. Burlina, and F. Alajaji. Image segmentation and labeling using the Polya urn model. *IEEE Trans. Image Proc.*, 8(9):1243–1253, 1999.

[12] A. Bavelas. Communication patterns in task-oriented groups. *Journal of Acoustical Soc. of America*, 22:725–730, 1950.

[13] M. Benaïm, I. Benjamini, J. Chen, and Y. Lima. A generalized Polya's urn with graph based interactions. *Random Structures & Algorithms*, 46, 2015.

[14] D. P. Bertsekas. *Nonlinear Programming.* Athena Scientific, 1 edition, 1995.

[15] M. Chen, S. Hsiau, and T. Yang. A new two-urn model. *Journal of Appl. Probab.*, 51:590–597, 2014.

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

[17] T. M. Cover and J. A. Thomas. *Elements of Information Theory.* Wiley, 2 edition, 2006.

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

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

[20] A. Fazeli and A. Jadbabaie. On consensus in a correlated model of network formation based on a Polya urn process. *2011 50th Conf. on Decision and Cont. and European Cont. Conf.*, 2011.

[21] W. Feller. *An Introduction to Probability Theory and its Applications*, volume 2. New York: Wiley, 2 edition, 1971.

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

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

[24] M. Hayhoe, F. Alajaji, and B. Gharesifard. A Polya contagion model for networks. *submitted, available at arXiv:1705.02239*, 2017.

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

[26] B. M. Hill, D. Lane, and W. Sudderth. A strong law for some generalized urn processes. *Annals of Math. Stats.*, 8:214–226, 1980.

[27] T. Huillet. On Polya-Friedman random walks. *Journ. Physics A: Math. and Theor.*, 41(50), 2008.

[28] S. Janson. Functional limit theorems for multitype branching processes and generalized Polya urns. *Stochastic Processes and their Applications*, 110:177–245, 2004.

[29] N. L. Johnson and S. Kotz. *Urn Models and Their Application.* Wiley, 1977.

[30] A. Khanafer, T. Basar, and B. Gharesifard. Stability of epidemic models over directed graphs: a positive systems approach. *Automatica*, 74:126–134, 2016.

[31] 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.

[32] S. Kotz, H. Mahmoud, and P. Robert. On generalized Polya urn models. *Stat. & Prob. Letters*, 49:163–173, 2000.

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

[34] 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.

[35] C. Nowzari, V. M. Preciado, and G. J. Pappas. Optimal resource allocation for control of networked epidemics. *IEEE Trans. Cont. Netw. Sys., to appear*, 2017.

[36] R. Pemantle. A time-dependent version of Polya's urn. *Journal of Theor. Probab.*, 3(4):627–637, 1990.

[37] R. Pemantle. A survey of random processes with reinforcement. *Probab. Surveys*, 4(0):1–79, 2007.

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

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

[40] E. Ramírez-Llanos and S. Martínez. Distributed and robust fair optimization applied to virus diffusion control. *IEEE Trans. Netw. Sci. and Eng.*, 4:41–54, 2017.

[41] A. Rarivoarimanana. *Unbalanced Urn Models and Applications.* PhD thesis, University of Cincinnati, 2014.

[42] H. Renlund. Generalized Polya urns via stochastic approximation. *arXiv preprint arXiv:1002.3716*, 2010.

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

[44] E. M. Rogers. *Diffusion of Innovations.* Simon and Schuster, 5 edition, 2003.

[45] B. M. Schreiber. Asymptotically stationary and related processes. In A. C. Krinik and R. J. Swift, editors, *Stochastic Processes and Functional Analysis: A Volume of Recent Advances in Honor of M. M. Rao.* CRC Press, 2004.

[46] Y. Wang, D. Chakrabarti, C. Wang, and C. Faloutsos. Epidemic spreading in real networks: an eigenvalue viewpoint. In *Proc. 22nd Intl. Symp. Reliable Dist Sys.*, pages 25–34, 2003.

[47] T. Zhu. *Nonlinear Polya Urn Models and Self-Organizing Processes*. PhD thesis, University of Pennsylvania, 2009.