Controlling Epidemics on Networks Using Stochastic Optimization Techniques

---

A

Dissertation

Presented to

the faculty of the School of Engineering and Applied Science

University of Virginia

---

in partial fulfillment

of the requirements for the degree

Doctor of Philosophy

by

Prathyush Sambaturu

May 2022

APPROVAL SHEET

This

Dissertation

is submitted in partial fulfillment of the requirements
for the degree of

Doctor of Philosophy

Author: Prathyush Sambaturu

This Dissertation has been read and approved by the examing committee:

Advisor: Anil Vullikanti

Advisor:

Committee Member: Madhav Marathe

Committee Member: B. Aditya Prakash

Committee Member: Abhijin Adiga

Committee Member: Jundong Li

Committee Member: Michael D. Porter

Committee Member:

Accepted for the School of Engineering and Applied Science:

Jennifer L. West, School of Engineering and Applied Science

May 2022

March 6, 2022

Abstract

We studied the problem of designing intervention strategies (e.g. vaccinations), under budget constraints, to minimize the spread of an epidemic outbreak. This is a challenging stochastic optimization problem in the context of the SIR epidemic model on a network. Previous approaches for this problem were either heuristics or approximation algorithms for restricted settings (e.g., transmission probability $p = 1$). We developed a bicriteria approximation algorithm, called SAAROUND, for the EpiControl problem, using techniques from stochastic optimization. Our algorithm provides empirical guarantees for solution quality in graphs of moderate size. We empirically evaluated our approach on various networks such as synthetic agent-based populations, random, and real-world collaboration networks. Our algorithm outperformed standard baseline heuristics (e.g., remove nodes with a high degree). Also, we showed that our approach obtained near-optimal interventions in practice.

The main bottleneck of the SAAROUND algorithm is using a solver to obtain a fractional optimal solution for the LP relaxation of the EpiControl problem. To overcome this bottleneck, we developed an approximation algorithm, adapting the Multiplicative Weights Update (MWU) method and the SAA technique, such that it bypasses the need to use a solver, to approximately solve the LP. We provided a memory-efficient version of this algorithm to scale this approach further, which allowed scaling to very large networks corresponding to state- and country-level populations.

Further, we considered a version of the EpiControl problem, where the sources might not be known precisely. In such a setting, a min-max objective, where the goal is to minimize the maximum expected outbreak size in any possible scenario, gives a more robust solution compared to the interventions considered for a single scenario. We developed rigorous approximation algorithms for this problem and evaluated its performance on different random graphs.

Finally, we considered the problem of extending our approach to control problems in other epidemic models that follow SIR class dynamics (e.g., SEI, SI). To this end, we developed a simple framework to extend our approach to such models. Particularly, we focused on the problem of designing group-scale interventions, to control the spread of invasive alien species (IAS), that affect crops, across a landscape. Our goal was to find a set of regions to intervene, satisfying budget constraints, such that the spread of IAS is minimized. We developed a bicriteria approximation algorithm for finding effective group-scale interventions for this problem and showed its performance guarantees. Further, we evaluated our algorithm on real-world networks and compared it with standard baselines.

Acknowledgements

This dissertation is a result of years long journey. I owe my gratitude to many who have been part of this journey. First and foremost, I thank my wife, Kamalika Ray, for her unconditional support without which this dissertation wouldn't have been possible. I thank my parents, Amma and Nannagaru, for their love and support. They have made my dream their own.

I thank my advisor, Anil Vullikanti, who has inspired me to become an independent researcher. He introduced me to the field of computational epidemiology, which is central to this dissertation. Our weekly meetings kept me motivated. I thank, Madhav Marathe, for being a mentor and the chair of my dissertation committee. He always encouraged me to aim high and provided valuable suggestions. I thank Ravi for being a mentor and a collaborator in multiple works. He was always approachable and I will cherish my conversations with him.

I am extremely glad to be part of a vibrant research institute such as the Biocomplexity Institute. I enjoyed my time working in the lab. I sincerely thank Abhijin Adiga for being an (unofficial) co-advisor to me. I had learnt a lot from my collaborations with him, particularly, related to designing the experimental setup and writing. I thank all my other collaborators at the Biocomplexity Institute: Srini, Bryan, Parantapa, and Mandy. I thank Aditya, Bijaya, Mahantesh, Marco, Ananth for their collaborations on works that are a central part of this dissertation.

I thank my friends Bargav, Sajal, Amogh, Swapna, Rounak, and Viresh.

I would like to thank all the funding agencies for supporting this work. This work has been partially supported by the following grants: NSF CRISP 2.0 Grant 1832587, NSF DIBBS Grant ACI-1443054, NSF EAGER Grant CMMI-1745207, NSF BIG DATA Grant IIS-1633028, and DTRA CNIMS (Contract HDTRA1-11-D-0016-0001).

Finally, I thank all the researchers in the world for pushing the boundaries of knowledge.

Contents

List of Figures

List of Tables

Dedication

To my family.

Chapter 1

Introduction

1.1 Motivation

Infectious diseases are responsible for millions of deaths and make many more people disabled each year, according to the World Health Organization (WHO) [72]. An *epidemic outbreak* is said to be a sudden rise in the number of cases of a disease-related illness within a community, population, or region. A *pandemic* is an epidemic that is widespread over multiple countries or even continents.

An epidemic can also lead to potential economic and social crises [100]. The ongoing COVID-19 pandemic only reinforced the need for studying computational problems such as modeling the epidemics, analyzing the spatio-temporal spread of an epidemic, and designing interventions, such as vaccinations, social distancing, and quarantining, to contain an outbreak.

During any large epidemic outbreak, public health agencies solve a variety of mathematical models to prepare guidelines and measures needed to contain the epidemic. These mathematical models can be broadly classified into (i) differential equation-based, and (ii) network or agent-based models. The differential equation-based models involve using a system of coupled differential equations to represent the dynamics [62, 95]. Typically, these models do not have any closed-

form solutions. However, when the system is small, they can be solved by brute force local search methods [62]. The second is stochastic agent-based models on social contact networks [27, 39, 59]. In these models, the complete mixing assumptions of differential equation models can be relaxed. These models are complex and more useful for modeling epidemics on large heterogeneous populations. These mathematical models are used extensively in studying the trade-off between the cost of interventions and the benefit of interventions (e.g., the number of people saved from infections). For instance, the CDC COVID-19 Scenario Hub [15,94] uses a variety of such models, both deterministic differential equations-based models [2, 62] and stochastic network-based models [17, 27, 35, 39, 59] in order to evaluate the benefits of different interventions, and finds the most effective ones.

Interventions such as quarantining infected individuals, closing schools, encouraging work from home, avoiding social gatherings, can help in controlling the epidemic, by reducing the transmissibility of the disease [22]. Vaccination and social distancing are the primary strategies for controlling the spread of epidemic outbreaks [6, 39, 57, 62, 71, 77–79, 98, 103, 104]. The production of vaccines is expensive and time-intensive. Therefore, there is always a shortage of vaccine supply. This makes the task of allocating vaccines under budget constraints very challenging.

Some of the objectives of interest in vaccine allocation are to minimize the expected outbreak size [14, 24], reduce the duration of the epidemic, lower the size of the peak, etc. The objective to lower the peak of the epidemic curve (i.e., a chart used to visualize the progression of an outbreak over time) is useful in cases, where the critical resources needed for patient care — such as hospital beds, ventilators, personal protective equipment (PPE), and so on — are scarce [13, 60, 64, 96]. However, this is a hard problem. The focus of this dissertation is designing interventions strategies intending to minimize the expected outbreak size.

At the start of every flu outbreak or during major pandemics, public health agencies seek to find *implementable* interventions to contain the outbreaks.There could be interventions that are effective, but might not be implementable, due to social and ethical issues [21, 47, 69]. For instance, targeted interventions, such as immunizing specific individuals, are not practical as they raise moral and social issues. Several implementation strategies [40, 58] are used to improve the adoption of interventions in a population.

Implementable strategies such as prioritizing immunization for people belonging to a particular age group [44] are typically used by public health agencies. Such strategies tend to be useful in the case of epidemics, where the risk of severe illness or death increases with the age of an individual [44]. These strategies are sub-optimal compared to targeted ones. Therefore, comparison with near-optimal strategies can help public health agencies to understand the cost incurred by the issues arising due to implementability [62]. As a result, there is a lot of interest in evaluating different kinds of interventions strategies [39, 57, 62, 95], and finding optimal interventions [62]. Such studies are useful in guiding policies, when there are shortages in vaccines, for instance, to decide the logistics of where and how the vaccines should be deployed [95].

1.2 Overview: Problems and Results

We consider the networked Susceptible-Infectious-Recovered (SIR) class models [59, 73] of disease spread for most of this dissertation. Assume we are given a social contact network, where nodes in the network represent people, and edges between any two nodes represent a connection over which an infection can be transmitted.

Interventions such as vaccinations and social distancing can be modeled as node removal and edge removal from the network, respectively. Figure 1.1 shows

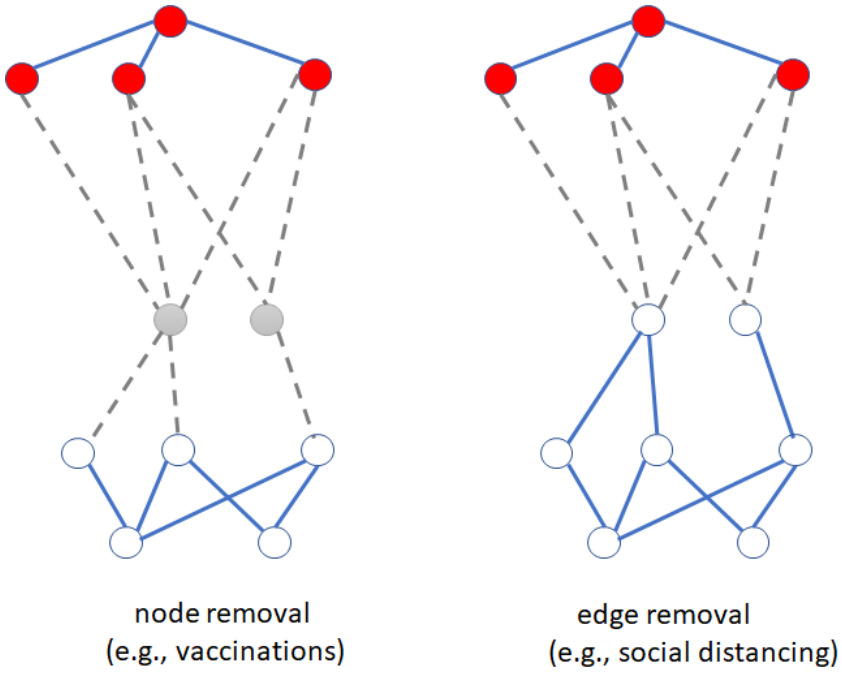


Figure 1.1: Types of interventions in the network-based epidemic models.

the node and edge removal interventions in the network-based epidemic models. The nodes in red correspond to infected people. The vaccinated nodes (or the nodes removed from the network) are represented in gray color. Note that, for each node that is removed, all the edges incident on it are removed as well. The node removal intervention models the assumption that a node, once vaccinated, will not get infected during the epidemic; consequently, it will not infect any of its neighbors. Whereas, in the case of edge removal (e.g. *social distancing or quarantining*), a person might avoid some contacts to reduce their risk of infection. But, they can still get the infection from the remaining active contacts.

This dissertation focuses on vaccination or node removal interventions. The goal is to find a subset of nodes in the network to remove (i.e., vaccinate), such that the expected epidemic outbreak size is minimized. In one chapter (Chapter 6) of this dissertation, we focus on group interventions, where a group represents a set of nodes. Intervening a group corresponds to removing all the nodes in the group from the contact network.

The contributions of this dissertation are approximation algorithms to find optimal intervention strategies in order to control epidemic spread. We designed our algorithms using techniques such as the sample average approximation (SAA) [54, 91] from stochastic optimization, linear programming and rounding [99], and multiplicative weights update (MWU) method [4]. We showed theoretical guarantees on the performance of these algorithms. Further, we evaluated the empirical performance of these algorithms on various networks.

The specific problems considered in this dissertation, the challenges they pose, and our contributions are presented in the rest of this section.

1.2.1 Algorithms to Minimize Expected Outbreak Size

Let us assume that we are given a contact network $G = (V, E)$, where V is the set of nodes (or people), and they are connected by an edge in E if they come into contact with each other. We assume a stochastic discrete-time SIR model on the contact network [59] for the epidemic spread, which can be summarized as follows: (i) each node in the network is in one of the following three states: Susceptible (S), Infectious (I), or Recovered (R), (ii) initially a small subset $S \subseteq V$ of nodes are infected, (iii) an infected node can infect each of its susceptible neighbors with probability p, referred to as *transmission probability*, and (iv) each infected node remains infected for $t_I > 0$ time steps and then moves to the recovered (R) state.

Problem 1: EpiControl. *Given a contact network $G = (V, E)$, sources of infection S (or source distribution), transmission probability p, and a budget B on number of interventions. The goal is to find a subset $\mathbf{X} \subseteq V$ of nodes in G to vaccinate, satisfying the given budget B on number of vaccines, such that the expected number of infections resulting from the SIR process is minimized.*

We consider non-adaptive interventions; i.e., all interventions are performed at

the beginning of the epidemic denoted by time $t = 0$. We assume that the vaccinations are immediately effective and have 100% efficacy. These assumptions, although not realistic, help to study this problem in a formal manner and achieve reasonable guarantees. The real problem of controlling epidemics is very complex with a lot of uncertainties. However, studying the solution characteristics of the nodes picked in the optimal solution can help identify "surrogates" for interventions in designing implementable strategies. Furthermore, the implementable interventions strategies available to public health agencies can be evaluated against the optimal solution. A precise formulation of the EpiControl problem is presented in Chapter 2.

Hardness of the EpiControl problem. The problem involves finding a subset of nodes from the network that optimizes a stochastic objective function (i.e., expected number of infections in G resulting from the SIR process). This is a very challenging stochastic optimization problem. This problem is **NP**-Hard, even for the case with transmission probability $p = 1$, shown in Hayrapetyan et al. 2005 [41]. The case with transmission probability $p = 1$ corresponds to a highly transmissible disease, where an infected node infects all of its susceptible neighbors, making this process deterministic. In this setting, it follows that any node that is reachable from the seed infected nodes (i.e., sources) will be infected. Therefore, all nodes in a component that have at least one seed node will be infected in the epidemic.

Previous approaches to this problem can be classified as follows:

(i) Heuristics: Many heuristics have been proposed based on local structure such as degree (selecting nodes with high degrees for vaccination), centrality, etc. These heuristics perform reasonably well in practice. However, they provide no guarantees for the objective of minimizing the expected number of infections in an epidemic.

(ii) Optimization of spectral properties: A fundamental result in epidemic

modeling is the characterization of an outbreak in terms of the spectral properties such as the first eigenvalue λ_1 of the adjacency matrix (also called the spectral radius) of the network, and the eigenvalues of the Laplacian [32, 76]. A property of spectral radius is that, the epidemic dies out if λ_1 is reduced to a value below a certain threshold (referred to as *epidemic threshold*). This result is used as a basis in many works to control the spread of epidemics [76–79, 81]. However, these methods do not provide any guarantees directly for this problem.

(iii) Algorithms with guarantees: This problem is **NP**-hard for the case with $p = 1$. The problem of designing pure approximation algorithms, for the case of EpiControl problem with transmission probability $p < 1$, is still open. Some of the previous works considered bicriteria approximation algorithms. Before the results in these works are summarized, a brief description of bicriteria approximate solutions is as follows.

Let us refer to the set of vaccinated nodes $\mathbf{X} \subseteq V$ as an *intervention set*. An intervention set is a (α, β)- bicriteria approximate solution for the EpiControl problem if (i) $\mathbf{X}$ is of size at most α times the given budget B, and (ii) the expected number of infections resulting from SIR process after removing nodes in $\mathbf{X}$ from contact network is at most β times that resulting from SIR process after removing nodes in $\mathbf{X}^*$, where $\mathbf{X}^*$ is an optimal solution for the EpiControl instance. A formal definition of bicriteria approximate solution appears in Chapter 2. As the problem is **NP**-hard even for the case with $p = 1$, earlier works such as Hayrapetyan et al. [41] and Eubank et al. [28], provide bicriteria approximation algorithms.

Our focus was to design intervention algorithms that obtain good approximation guarantees on both the criteria of EpiControl problem: objective (expected number of infections) and budget. More importantly, to obtain empirical approximation factors for the given problem instance. This is very useful to evaluate the performance of other standard interventions strategies (e.g. degree) deployed

by public health agencies [17].

Our contributions to this problem are briefly described below. Since the problem is **NP**-hard even for the case with $p = 1$, and pure approximations are still open, we too considered designing bicriteria approximation algorithms.

saaRound **Algorithm.** [83] We first designed a bicriteria approximation algorithm for the EpiControl problem using the sample average approximation (SAA) technique [54, 85, 86] from stochastic optimization, linear programming and rounding techniques.

The main idea of this algorithm is to generate M sampled outcomes of the SIR process for a sufficiently large value of M. Then, solve a deterministic problem on the M samples, which involves finding a solution (i.e., subset $\mathbf{X} \subseteq V$ of nodes to vaccinate) that minimizes the average number of infections in these M sampled outcomes. This problem is very hard to solve with efficient algorithms. To see this, consider the case with $p = 1$, there is only one unique sample since the process is deterministic. The resulting version of the problem is **NP**-hard [41]. Therefore, we considered linear program (LP) relaxation of the problem. However, the optimal solution to the LP relaxation is a fractional solution. We designed a randomized rounding procedure to obtain an intervention set from this optimal fractional solution. We showed in our analysis that the intervention set thus obtained has bicriteria approximation guarantees for the EpiControl problem instance.

Scalable Algorithms. In recent works, agent-based models on larger populations are used, for instance, Chen at al. [17], to study a model for Virginia in the United States. They considered a degree-based strategy for interventions. Also, the state and the national-level agent-based models are also used in the CDC modeling hub [15]. This motivates scaling the saaRound approach to state- and country-level populations.

However, the saaRound algorithm only scales up to networks with 10^5 nodes,

typically the size of a county in the United States. The main bottleneck of this approach is that it uses LP solvers for the linear program relaxation of the problem.

We designed fast and scalable algorithms, by adapting the MWU method [4], to approximately solve the LP relaxation of the EpiControl problem. The Multiplicative Weights Update method maintains a distribution over a set of values and then iteratively updates these weights based on a multiplicative update rule [5]. This approach is well studied and used in various fields and for various problems [31,75]. In our approach, which is an adaptation of Fleischer, 2000 [30], the variables are initialized with small values and are updated in each iteration, based on certain criteria, until all the constraints of our problem are satisfied.

Our main contributions to EpiControl problem can be summarized as follows:

- We designed saaRound algorithm for the EpiControl problem. We showed that it obtains $(O(\log(nN)),\ 6)$-bicriteria approximate solutions for the problem with high probability. Here, n is the number of nodes in the contact network and N is the maximum number of paths from a source to any node in any sampled outcome. We note that in the worst-case, the guarantees for the budget criteria can be $O(n)$. However, the empirical approximation factor for budget is a very small value in practice.

- saaRound provides the empirical approximation factor for the problem instance, relative to an optimum.

- We performed a detailed experimental evaluation of saaRound algorithm on various real-world, random, and synthetic population networks. Our results showed that saaRound algorithm has near-optimal performance in practice. It significantly outperformed the standard baselines for this problem.

- We designed a scalable algorithm, referred to as MwuRound, using the MWU method with SAA technique and randomized rounding. We showed that this algorithm provides $(O(\log(nN)),\ 6(1+4\epsilon))$- bicriteria approximate solution for the EpiControl, where ϵ is an error parameter.

- We showed that MwuRound algorithm has near-optimal guarantees for EpiControl problem, similar to that of saaRound algorithm, in practice. Furthermore, we showed that MwuRound algorithm is able to run for large networks with millions of nodes without any memory issues. However, this algorithm is slow on large networks.

- Finally, we designed a fast, scalable, and memory-efficient version of MwuRound algorithm, which obtains a feasible solution to the problem. We showed that this algorithm scales well to large networks corresponding to country-size populations. More importantly, we showed that this algorithm has good performance guarantees in practice.

1.2.2 Robust Intervention Algorithms

The EpiControl problem assumes that the source set or their distribution, and the transmission probability p are known. However, many components of the SIR epidemic model, such as transmission probability, p, source distribution, or seed source set, might not be known precisely. The seed source set (or the source distribution) are called the initial conditions of an epidemic. In such settings, where the initial conditions are not known precisely, a *min-max* type objective, where the goal is to minimize the maximum expected number of infections in any given scenario, is more suitable than tailoring the interventions to a single scenario. The use of a min-max objective to handle uncertainty is well motivated and has a solid foundation in the field of stochastic optimization [42, 87].

Problem 2: MinMaxEpiControl. *Given a contact network G, a set $\mathcal{I}$ of*

possible scenarios, where each scenario corresponds to a different set of sources. Let budget B denote the number of available vaccinations. The goal is to find a subset $\mathbf{X} \subseteq V$ of nodes in G to vaccinate, satisfying the given budget B, such that the maximum expected number of infections resulting from any scenario from $\mathcal{I}$ is minimized.

Summary of contributions [84] related to MinMaxEpiControl are as follows:

- We formalized the MinMaxEpiControl problem, to design robust interventions, in the case where the initial conditions of an epidemic are unknown.

- We designed an approximation algorithm called mmRound and showed its performance guarantees. For the version of MinMaxEpiControl problem with $p < 1$, we adapted the SAA and LP rourounding based approach used in saaRound algorithm. Specifically, for the case with transmission probability $p = 1$, mmRound algorithm was modified to use a graph separator subroutine, which gave a better approximation factor (logarithmic in the number of nodes) on the budget than shown for the $p < 1$ case.

- Empirically, we showed that the solutions to the min-max objective, are very different from those picked for a specific source distribution.

1.2.3 Extensions to Other Epidemic Models

As discussed in the previous sections, the problem of designing intervention strategies, under budget constraints, is challenging even for a simple SIR epidemic model. The two problems considered so far make a lot of assumptions that simplify the epidemic model. However, in reality, epidemic processes tend to be very complex. So, the questions that arise are: (i) Can we adapt our saaRound approach to any other model in the SIR class dynamics (e.g. Susceptible-Infectious-

Exposed (SEI))? (ii) Is it possible to apply our techniques to epidemic control problems that assume complex epidemic models that follow SIR class dynamics?

These questions led us to consider the problem of designing interventions to control the spread of invasive alien species (IAS) (e.g., *Tuta absoluta [10]*) across a landscape. McNitt et al. [61] modeled the IAS spread as a multi-scale epidemiological process named MultiPath, whose dynamics follow a discrete-time Susceptible-Exposed-Infectious (SEI) [59] process. The study region can be represented by a spatial network, where the nodes correspond to cells (i.e., crops). The network considered in MultiPath model is a directed edge-labeled and edge-weighted temporal network. Groups correspond to a set of spatially contiguous cells. Group-scale interventions are more practical for this problem.

The SEI model belongs to the SIR class of dynamics. The key difference in the SEI model is that, once a node is infected it enters the exposed state E and remains in that state for a certain period of time before moving to the infectious state I. Only when it is in the state I, it can infect its susceptible neighbors. The other models in SIR class dynamics include Susceptible-Infectious (SI), Susceptible-Infectious-Susceptible (SIS), and Susceptible-Exposed-Infectious-Recovered (SEIR) epidemic models [26, 59].

Problem 3: IAScontrol problem. *The goal is to find a subset of regions (or groups) to intervene, under budget constraints, such that the expected number of nodes infected*[1] *at a time horizon T due to SEI process is minimized.*

The steady-state for an SEI process is when all the nodes reachable from source infections are exposed. Therefore, this problem is meaningful only in a setting with a finite time horizon T. A formal description of IAScontrol problem appears in Chapter 2. A summary of our contributions related to this problem and extensions to other models is presented below.

[1] The formal definition of this problem considers the objective of minimizing expected number of nodes exposed at time horizon T, since the nodes exposed at time-step T will eventually get infected

- We showed that the MultiPath epidemic for a finite time horizon can be reduced to an SIR process on the corresponding time-expanded network.

- We showed that the IAScontrol problem is **NP**-complete even when G is a tree. Further, we showed that a variation of this problem where the goal is to minimize the cost of the interventions to ensure that the expected number of infections is bounded is very hard to approximate.

- We designed a bicriteria approximation algorithm SpreadBlocking for the IAScontrol problem and showed guarantees on its performance.

- We studied the performance of SpreadBlocking on real-world networks and note that its performance is superior to the baselines for this problem. Further, we showed that SpreadBlocking has good performance guarantees in practice.

- We presented a framework to extend our saaRound approach to other complex epidemic models that follow SIR-class dynamics.

1.2.4 Summary and Takeaways

The work in this dissertation broadly focuses on controlling SIR class epidemics on networks. These are hard problems, as they involve selecting a subset of nodes from the network to optimize a stochastic objective function. Our research shows the following.

(i) **SAA+LP+Rounding**: The sample average approximation (SAA) technique, along with LP and rounding, is very useful in designing algorithms that provide reasonable guarantees for such problems. Particularly, using the SAA technique reduces the problem from a stochastic optimization to a deterministic one on samples. Then, the LP and rounding techniques are used to solve the deterministic problem on samples. More importantly, we note that these

algorithms are capable of obtaining near-optimal solutions in practice. A key limitation of this approach is scalability, as solving the LP exactly using a solver, is a bottleneck.

(ii) **SAA+MWU+Rounding**: The SAA technique combined with the MWU method and rounding, aids in designing scalable algorithms for such stochastic optimization problems. The MWU method allows to approximately solve the LP without losing much of the guarantees. Therefore, this combination of techniques overcomes the limitations of SAA+LP based approach.

(iii) **Framework for other epidemic models**: Our approach can be extended to control problems in complex epidemic models (e.g. MultiPath), whose dynamics follow SIR class models, using a simple framework (Chapter 6):

(Step 1.) Represent the dynamics of the complex epidemic model on a given network as a SIR process on an auxiliary network. This can be achieved, for certain models, by the notion of auxiliary graphs such as time-expanded networks.

(Step 2.) Solve the problem of the designing optimal interventions corresponding to the SIR process on the auxiliary graph, which is generated in Step 1. Adapt the saaRound or MwuRound approach to solve this problem.

1.3 Thesis Organization

The organization of the dissertation is as follows:

Chapter 2 introduces the necessary notations and preliminaries to formally define each problem considered in this dissertation. The rest of this chapter provides a brief background on the material needed to understand this dissertation.

Chapter 3 presents an overview of the literature related to the works in this dissertation. In particular, it gives an overview related to different mathematical models considered for the intervention problems, and the related work on

designing interventions for each of these models.

Chapter 4 presents the intuition behind the saaRound algorithm and its analysis. Next, it presents an experimental evaluation of the empirical performance of saaRound algorithm, and the characteristics of the solutions obtained by it.

Chapter 5 presents the mmRound algorithm and the extensions of this approach to $p < 1$ and the case of random sources. Chapter 5 ends with experimental results on random graphs.

Chapter 6 presents extensions of the saaRound approach to other epidemic models that follow SIR class dynamics. First, it presents an approximation algorithm for the IAScontrol problem, designed by adapting the saaRound approach, and performance guarantees. Next, it presents an experimental evaluation of this algorithm on real-world networks and a comparison to standard baselines. Further, this chapter also presents a framework to extend our saaRound approach to other epidemic models based on the ideas used to solve IAScontrol problem.

Chapter 7 presents MwuRound algorithm, which overcomes the bottleneck in saaRound, for EpiControl problem. This chapter presents the performance guarantees of this algorithm. Further, a memory-efficient and scalable version of this algorithm is presented. Chapter 7 ends with an experimental evaluation of these algorithms.

Chapter 8 provides the conclusions and a short list of open questions.

Chapter 2

Preliminaries and Problem Statements

In this chapter, we introduce the notation and the formal statements of the problems considered in this dissertation. Further, necessary background to understand the material in the following chapters is presented.

2.1 SIR Epidemic Model on Networks

Contact network. Let $G = (V, E)$ be a contact network where V denotes the set of people (also referred to as nodes) and $e = (u, v) \in E$ if nodes $u, v \in V$ come into direct contact, which can allow a disease to spread. Let $n = |V|$ and $m = |E|$ denote the number of nodes and edges in the contact network G respectively.

Disease model. We assume a simple SIR model of disease spread [59] on networks. Each node in the network is in one of the following three states:
(i) Susceptible (S): nodes that are not yet infected but are susceptible to infection,
(ii) Infectious (I): nodes that are infected and can potentially spread the infection to those that come into contact with them,
(iii) Recovered (R): nodes that were infected and recovered from the infection.

In some cases, this state may also include deceased individuals, who are removed from the process.

To simplify the exposition, we assume that the epidemic starts at a set of externally infected nodes denoted by S. The set of nodes in S are referred to as *seeds* or *sources* of the infection. We assume discrete-time, where at any time $t \in \{0, 1, ..., \tau\}$, all the nodes in the network are in one of the three states: S, I, and R. At the time $t = 0$, the sources are in state I, and all the other nodes are in state S. At any time step $t > 0$, the disease spreads from an infected node u to each of its susceptible neighbors with a probability p, referred to as transmission probability. An infected node recovers in the next time step.

The results in this dissertation also hold for the case where each node u has a probability s_u to be infected initially. This is denoted as source distribution $\mathbf{s}$. Necessary modifications to our algorithms, for this case, are presented in sufficient detail.

The SIR model generalizes the well studied *independent cascades* model [49].

There are lots of variations of the SIR model, such as:

1. Each edge has a probability $p(u, v)$ for transmission instead of a uniform probability p for all edges. This makes sense in situations where some contacts have a high rate (e.g., working in the same place) of spreading the infection.

2. Variable infectious state duration, where a node u remains in state I for $t_I(u)$ time steps.

The SIR class of epidemic models includes (but is not limited to) the following models:

(i) Susceptible-Infectious (SI): A node once infected, remains in state I throughout the rest of the process.

(iii) Susceptible-Infectious-Susceptible (SIS): Each infected node returns back to the susceptible state S after a certain period of time. This models the epidemics in which reinfections are possible.

(iii) Susceptible-Exposed-Infectious (SEI): This is similar to SI model with a key change that an infected node first enters the exposed state E after getting infected. It stays in that state for a certain period of time before moving to state I. The length of the period during which an infected node stays in an exposed state is referred to as the latency period.

(iv) Susceptible-Exposed-Infectious-Recovered (SEIR): This is similar to the SIR model, except that an infected node first enters the exposed state E after getting infected. Then, the node stays in state E for a certain period of time after which it moves to state I.

2.2 EpiControl Problem

First, we present the notation necessary for the EpiControl problem. Then, we formally define the problem.

A stochastic outcome from SIR process. Given a contact network $G = (V, E)$, a source set of infections S, and a transmission probability p. Let us denote a stochastic outcome from the SIR process by $H^{(sir)} =< I(0), \ldots, I(\tau), E' >$, where

(a) $I(t)$ denotes the set of nodes that are infected at time t,

(b) $I(0)$ denotes the source nodes; and

(c) $E' \subseteq E$ is the random subset of edges on which the infection spread.

The number of infections in a stochastic outcome from the SIR process is denoted by $\#\text{infections}(G, S, p)$. This is a random variable. The expected number of infections in G resulting from the SIR process is given by

$$\mathbb{E}[\#\text{infections}(G, S, p)] = \mathbb{E}_{H^{(sir)}}[\sum_{t} |I(t)|] \tag{2.1}$$

where, the summation on the right hand side of equation (2.1) corresponds to the total number of infections over all the time steps.

Interventions and objective.

Let x_u for $u \in V$ be an indicator variable defined as

$$x_u = \begin{cases} 1 & \text{if node } u \text{ is vaccinated} \\ 0 & \text{otherwise} \end{cases} \tag{2.2}$$

Then, $\mathbf{X} = \{u : x_u = 1, u \in V\}$ denotes the set of vaccinated nodes called the *intervention set*. Let B denote the number of vaccines available, also referred to as *budget*. The budget constraint then is given by $|\mathbf{X}| \leq B$.

We consider non-adaptive intervention setting, which means that all the interventions are determined ahead of time at the beginning of the epidemic, say $t = 0$. Further, we assume that the vaccines have 100% efficacy and are immediately effective.

We extend the earlier notation, and define $H^{(sir)}(\mathbf{X}) =< I(0), \ldots, I(\tau), E' >$ to be a stochastic outcome from the SIR process when $\mathbf{X}$ is the *intervention set*, where the interventions performed at $t = 0$. This would mean that none of the nodes in $\mathbf{X}$ are part of any $I(t)$.

Let $\#\text{infections}(G, S, p, \mathbf{X})$ denote the number of infections in a stochastic outcome of SIR process, when $\mathbf{X}$ is the *intervention set*. The expected number of infections is denoted by $\mathbb{E}[\#\text{infections}(G, S, p, \mathbf{X})]$. When the context is clear, we omit G, S, and p from these definitions as follows: $\#\text{infections}(\mathbf{X})$ and $\mathbb{E}[\#\text{infections}(\mathbf{X})]$.

Example. Figure 2.1 shows the SIR model and the definitions of above quantities on a contact network, $G = (V, E)$, which is shown on the left. Here, $V = \{A, B, C, D, E, F\}$ is a set of people, and connections (or edges) are shown as solid lines. It is assumed that node A is initially infected; i.e., it is a seed infected node, and node C is vaccinated. This corresponds to $S = \{A\}$ and the *intervention set* $\mathbf{X} = \{C\}$. The four possible stochastic outcomes in the SIR

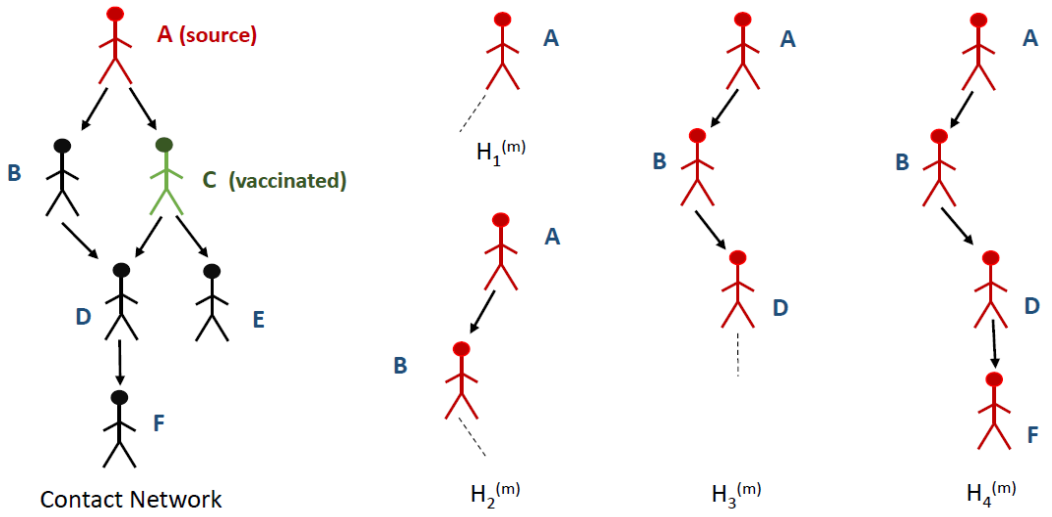


Figure 2.1: Example illustrating the SIR model and the notation of EpiControl problem

model, $H_1^{(sir)}, H_2^{(sir)}, H_3^{(sir)}$, and $H_4^{(sir)}$ are shown on the right.

Recall that, in the SIR model, the disease spreads from an infected node to each of its susceptible neighbors with probability p, and the infection does not spread with probability $1-p$.

Therefore, we have exactly four possible stochastic outcomes in this case, $H_1^{(sir)}, H_2^{(sir)}, H_3^{(sir)}$ and $H_4^{(sir)}$, each of which occur with probabilities $1-p$, $p(1-p)$, $p^2(1-p)$, and p^3, respectively. Then, the expected number of infections is given by the following equation.

$$\mathbb{E}[\#\text{infections}(\mathbf{X})] = 1 \cdot (1-p) + 2 \cdot p(1-p) + 3 \cdot p^2(1-p) + 4 \cdot p^3$$

We have set up all the necessary notation to define the EpiControl problem. Now, a formal definition of the problem is as follows:

Definition 1. EpiControl problem.

Instance. Given a contact network $G = (V, E)$, sources of infection S, transmission probability p, and a budget B on the number of interventions.

Goal. To find an intervention set $\mathbf{X} \subseteq V$ such that $|\mathbf{X}| \leq B$ and the expected

number of infections $\mathbb{E}[\#\text{infections}(G, S, p, \mathbf{X})]$ is minimized.

Note. Our results (in Chapters 4 and 7) extend to the case of random sources with a source distribution $\mathbf{s}$, where each node u has a probability s_u to be a seed infection.

Most of the algorithms in this dissertation provide bicriteria approximation guarantees, i.e., guarantees on both the expected number of infections objective and the violation of budget. So, we formally define a bicriteria approximate solution (and algorithm) for the EpiControl problem. This definition can be generalized to any optimization problem including the MinMaxEpiControl and IAScontrol problems.

Definition 2. Bicriteria approximate solution for EpiControl problem.
We refer to an *intervention set* $\mathbf{X}$ as a (α, β) - approximate solution for a given instance of the EpiControl problem if:
(1) $|\mathbf{X}| \leq \alpha B$, and
(2) $\mathbb{E}[\#\text{infections}(\mathbf{X})] \leq \beta\ \mathbb{E}[\#\text{infections}(\mathbf{X}^*)]$, where $\mathbf{X}^*$ is an optimal solution for the instance of EpiControl problem.

We say that an algorithm is a (α, β)-approximation algorithm for the EpiControl problem, if it gives an (α, β)-approximate solution for any instance of the problem.

2.3 MinMaxEpiControl

We assume a networked SIR model (described in Section 2.1). The disease spread depends crucially on the initial conditions (i.e., sources). Let S denote the set of nodes at which the outbreak starts; this could be an *arbitrary* subset of nodes, or could also be a distribution (e.g., uniform, or degree biased).

Let $\mathcal{I}$ indicate a set of scenarios corresponding to different initial conditions. This models the setting in which any one of these scenarios is possible, but we don't precisely know which. Our goal is to find a solution that works well for all of them. We will generally assume each $S \in \mathcal{I}$ corresponds to a small set of deterministic or probabilistically chosen subset of nodes.

Figure 2.2 shows two different scenarios, each with a different source set of infections. The contact network $G = (V, E)$ is shown on the left, where $V = \{a, b, c, d, e\}$ represents nodes and are shown in circles. The connections or edges are shown as solid lines. In the first scenario (top), node a is initially infected (i.e., $S = \{a\}$), and node b is vaccinated. The five subgraphs O_1, O_2, O_3, O_4, O_5, shown on the top-right, are possible stochastic outcomes in the SIR model. The probability of occurrence for each of these outcomes is indicated below the outcome. This scenario is the same as the one shown in Figure 2.1. In the second scenario (bottom), node c is initially infected (i.e., $S = \{c\}$), and node b is vaccinated. In this case, there are eight outcomes $Q_1, \ldots, Q_8$.

Interventions. We assume that the vaccine is perfect (i.e., 100% efficacy). This means that a vaccinated node does not get infected. We will first consider the *one-stage* vaccination strategy, which involves picking a set of nodes $\mathbf{X}$ to vaccinate at the start of the outbreak. Let B denote the budget for the number of vaccines available.

We will also consider a *two-stage* intervention problem, where vaccines are allocated in two stages: (i) at the beginning, $t = 0$ and (ii) at time $t = T$. Let the sets $\mathbf{X}_0$ and $\mathbf{X}_T$ denote the nodes picked for vaccination at $t = 0$ and $t = T$, respectively. The sets $\mathbf{X}_0$ and $\mathbf{X}_T$ will be referred to as *intervention sets* at times 0 and T, respectively.

Objective. Let $\texttt{numinf}(G, S, p, \mathbf{X})$ denote the number of infections in G, in a stochastic outcome from SIR process, when the outbreak starts at $S \in \mathcal{I}$, and $\mathbf{X}$ denotes the interventions. Notice that, we use a different notation for the number

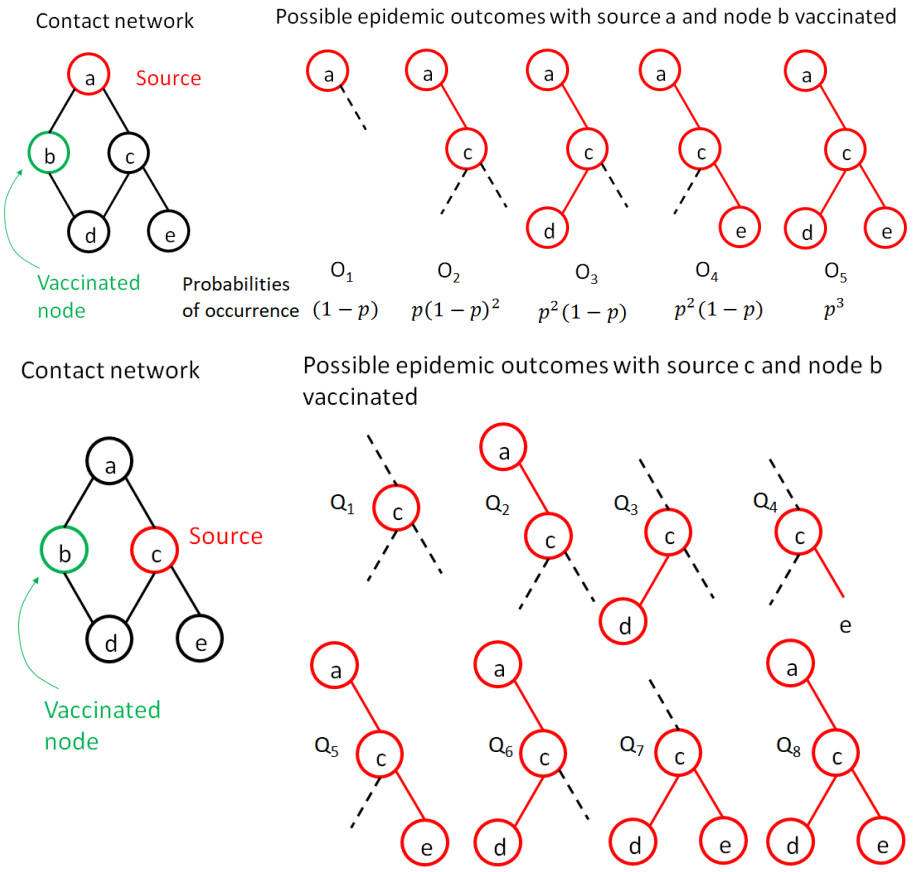


Figure 2.2: SIR outcomes for two different scenarios.

of infections in the SIR process, in the case of this problem instead of the one used for the EpiControl problem. This is done deliberately to distinguish the assumption made in this problem: the sources (or source distribution) are not fixed.

Let $\mathbb{E}[\texttt{numinf}(G, S, p, \mathbf{X})]$ denote the expected number of infections in G when the outbreak starts at $S \in \mathcal{I}$. We omit G and p from this notation when the context is clear.

Example. Figures 2.2 shows two scenarios, one with node a as the source, and the other with node c as the source. The set $\mathbf{X} = \{b\}$ denotes the intervention set.

Then, the expected number of infections in the first scenario is given by

$$\mathbb{E}[\texttt{numinf}(S = \{a\},\ \mathbf{X} = \{b\}]) = 1 \cdot (1-p) + 2 \cdot p(1-p)^2 + 3 \cdot 2p^2(1-p) + 4 \cdot p^3$$

On the other hand, when $S = \{c\}$ (as shown in the bottom scenario in Figure 2.2), the expected number of infections is given by the following equation.

$$\mathbb{E}[\texttt{numinf}(S = \{c\},\ \mathbf{X} = \{b\}]) = 1 \cdot (1-p)^3 + 2 \cdot 3p(1-p)^2 + 3 \cdot 3p^2(1-p) + 4 \cdot p^3$$

The problem of designing robust interventions for a set of scenarios $\mathcal{I}$ is formally defined below.

Definition 3. Min-Max vaccination problem (MinMaxEpiControl).
Instance. Given a contact network $G = (V, E)$, a set $\mathcal{I}$ of possible scenarios where each scenario $S \in \mathcal{I}$ corresponds to a set of initial sources of infection, and budget B on the vaccines
Goal. To find a set $\mathbf{X}$ of nodes to vaccinate such that $\max_{S \in \mathcal{I}} \mathbb{E}[\texttt{numinf}(S, \mathbf{X})]$ is minimized, and $|\mathbf{X}| \leq B$.

2.4 IAScontrol problem

In this section, we introduce the problem of controlling the spread of invasive alien species (e.g. pests) using group-scale interventions.

The MultiPath model for IAS spread. We will briefly describe the model developed in McNitt et al. [61], referred to as the MultiPath model. For a detailed description of this model, we refer to [61, 90].

The region of interest is divided into cells (e.g. crops). The cells represent the nodes in a spatial network $G = (V, E)$. A group of spatially contiguous nodes

(e.g., localities) represent the regions of major supply of host crops and demand. Many nodes in this spatial network do not belong to any locality.

We consider group-scale interventions, where an intervention corresponds to removing all the nodes in the group, selected for intervention, from the network. Nodes that do not belong to any group (or locality) are not candidates for group-scale interventions.

Let $\mathcal{Q}$ be a collection of k disjoint subsets of the vertex set V, where each subset represents a group. Let $g(v) \in \mathcal{Q}$ denote the group to which node v belongs.

This model considers three pathways of spread:

(i) Self-mediated dispersal: This represents the diffusion from a crop (or cell) to its neighboring crops.

(ii) Local human-mediated dispersal: This represents the diffusion within a group, such as farmer-market interactions.

(iii) Long-distance dispersal: This represents the diffusion from cells of one group to those in another group, typically via trade.

The diffusion model is a discrete-time SEI process. A node transitions from exposed state **E** to infectious state **I** after ℓ time steps, where ℓ is the latency period.

The transition from **S** to **E** is described as follows. A node has two periodic time-varying attributes called (i) suitability $\epsilon(v, t)$ for the pest (IAS) establishment and (ii) infectivity $\rho(v, t)$.

The probability that a node can be infected through a pathway is modeled as a negative exponential function of infectivity and pathway parameters. These probabilities are modeled as edge weights between the two nodes.

The probability that node v is infected by its neighbor v' within its Moore

neighborhood (short-distance dispersal) is given by

$$w(v', v, \lambda_s, t) = \epsilon(v, t)\big(1 - \exp(-\alpha_s \rho(v', t))\big),$$

where λ_s is the edge label corresponding to the pathway and α_s is a tunable pathway parameter.

If the two nodes v and v' are within a group, the probability of within group transmission (human-mediated dispersal) from v' to v is given by

$$w(v', v, \lambda_\ell, t) = \epsilon(v, t)\big(1 - \exp(-\alpha_\ell \rho(v', t))\big),$$

where λ_ℓ is the pathway label and α_ℓ is the pathway parameter.

For the group-to-group transmission, a directed flow network is defined with groups as nodes and the edge weight for the edge from Q_i to Q_j denoted by F_{ij}. Suppose $g(v) = Q_j$ and $g(v') = Q_i$, then the probability that v is infected by v' through this pathway is given by

$$w(v', v, \lambda_{\ell d}, t) = \epsilon(v, t)\big(1 - \exp(-\alpha_{\ell d} F_{ij} \rho(v', t))\big),$$

where $\lambda_{\ell d}$ is the pathway label and $\alpha_{\ell d}$ is the pathway parameter.

The complete details of network construction are in McNitt et al. [61]. Now, we present the notation necessary to define our problem.

Notation and Problem Statement. Let $G = (V, E)$ be a temporal edge-weighted and edge-labeled directed graph. Let the weight of an edge $(u, v, \lambda, t) \in E$ at a discrete time step t and label λ be denoted by $w(u, v, \lambda, t)$. Let $\mathbf{s}$ denote the seed set of infections.

Let $\mathcal{Q} = \{Q_1, Q_2, \cdots, Q_k\}$ be a collection of k disjoint subsets of the vertex set, where each Q_j is a group.

Intervening at a group means removing all the nodes in the group. The

interventions are performed τ_{d} time steps after the source is known, so τ_{d} is referred to as *intervention delay*. The interventions are non-adaptive (as in the earlier problems). This means that the decision to intervene is not made by observing the system state at some time step in range $[1, \tau_{\mathrm{d}} - 1]$. Instead, this decision is based on the expected state of the system at τ_{d}.

Suppose $V' \subseteq V$ is the set of nodes intervened at τ_{d}, then, let $\inf_{\mathrm{T}}(G, \mathbf{s}, \tau_{\mathrm{d}}, V')$ (we can drop $G, \mathbf{s}, \tau_{\mathrm{d}}$ when context is clear) denote the expected number of nodes exposed at a time horizon T due to SEI diffusion with source nodes $\mathbf{s}$ when nodes in V' are intervened at time τ_{d}.

Note that the steady-state for an SEI process is when all the nodes reachable from seed infections $\mathbf{s}$ become exposed (and then infected). Therefore, the intervention problem is relevant only when the time horizon T is finite.

The IAScontrol problem is formally defined as follows.

Definition 4. IAScontrol problem

Instance. Given a temporal edge-weighted and edge-labeled directed pathway network $G(V, E)$, a partition of the vertex set V into k groups $\mathcal{Q}$ with a cost (of intervention) c_q for each $Q_q \in \mathcal{Q}$, source nodes $\mathbf{s} \subseteq V$, SEI diffusion process on G with transmission probabilities equal to the edge weights, budget B, intervention delay τ_{d} and time horizon T.
Goal. Find a set of groups $\mathcal{Q}^* \subseteq \mathcal{Q}$ to intervene such that $\sum_{Q_q \in \mathcal{Q}^*} c_q \leq B$ and the expected number of infections $\inf_{\mathrm{T}}(G, \mathbf{s}, \tau_{\mathrm{d}}, \{v \mid g(v) \in \mathcal{Q}^*\})$ is minimized.

In this dissertation, we will primarily focus on the unweighted version of IAScontrol, where $c_q = 1$ for all $q \in \mathcal{Q}$.

An alternative form of the problem called IAScontrolMinBudget problem has a goal to minimize the number of groups intervened such that the expected number of infections is upper bound by some K. This can be stated as follows. Given a target bound K on the number of infections, choose $\mathcal{Q}^*$ so that

$\inf_{\mathrm{T}}(G, \mathbf{s}, \tau_{\mathrm{d}}, \{v \mid g(v) \in \mathcal{Q}^*\}) \leq K$, and $\sum_{q \in \mathcal{Q}^*} c_q$ is minimized.

We do not provide any algorithms for the IAScontrolMinBudget problem, but just discuss its hardness.

2.5 Technical Background

Stochastic Optimization. Stochastic Optimization (SO), also known as Stochastic Programming, involves methods to maximize or minimize an objective function (i.e., making optimal decisions) when randomness is present. We refer to [11] for interested readers. Sample average approximation (SAA) technique is a standard and natural approach used to solve stochastic optimization problems [54,91]. The basic idea of SAA is that solving the problem on the samples is enough to get a "good" solution for the stochastic objective. Therefore, we solve the deterministic problem on the samples to achieve a solution.

Tail Bounds. Analysis of algorithmic guarantees often involves bounding the probability that a random variable deviates far from its mean. The weakest tail bounds are Markov's and Chebyshev's inequalities [45]. This is because Markov's and Chebyshev's inequality converge linearly and quadratically, respectively. A much more powerful tail bound referred to as Chernoff bounds [19] are derived using Markov's inequality on the moment generating functions of a random variable. An interested reader can refer to [68] for a thorough background in these concepts.

We use the following version of the Chernoff bound in our analysis (Chapter 4).

Theorem 5. *(Theorem 1.1 of [23]) Let $Z = \sum_{i=1}^{n} Z_i$, where Z_i are independently distributed random variables in $[0, 1]$. Then, for any $\epsilon \in (0, 1)$, we have $\Pr[Z \notin [(1-\epsilon)E[Z], (1+\epsilon)E[Z]]] \leq 2exp(-\epsilon^2 E[Z]/3)$. Also, for any $t > 2eE[Z]$, $\Pr[Z > t] \leq 2^{-t}$.*

Chapter 3

Related Work

There is a huge amount of literature on interventions for epidemic models. First, we will present background on epidemic models and then discuss related work on interventions.

3.1 Mathematical Models for Epidemiology

Epidemic outbreaks shaped the course of human history, causing the fall of empires and collapse of civilizations [74]. The Black Death outbreak in Europe in 1348 resulted in over 25 million deaths. The cocoliztli epidemics or The Great Pestilence in the 16th century caused over 13 million deaths [1], decimating the native population in present-day Mexico. The "Spanish" Influenza pandemic infected about an estimated 500 million and caused over 50 million deaths during 1918-1919. The ongoing COVID-19 pandemic has already infected over 250 million people (as of November 2021), as the threat of a new variant is looming.

The first known use of mathematical models in epidemiology is attributed to Bernoulli, who developed a mathematical model to show that inoculation against the smallpox virus increased the life expectancy at birth by three years [12]. In 1911, Ross developed the first differential equation model of malaria [89]. Following this, Kermack and McKendrick in their seminal works [50–52] founded

the deterministic compartmental modeling using ordinary differential equations (ODEs) based on mass-action model. In these models, the population is divided into compartments or subgroups (e.g., disease states). The basic reproductive number denoted by R_0 is defined as the expected number of secondary infections resulting from a single infective individual into an entirely susceptible population. The value of R_0 is used to determine if an epidemic will occur or not [50]: if $R_0 < 1$, the epidemic will die out; otherwise there will be an epidemic. Therefore, estimating R_0 value is very useful to public health policymakers in planning their response to an outbreak or evaluating the effectiveness of the policies in place [46, 80].

There are two broad classes of epidemiological models: (i) differential equations-based models, and (ii) network-based models. Differential equation-based models are used to predict the trajectory of the outbreaks [43, 48], which helps in evaluating the public health policies used to control the spread of outbreaks. These coupled differential equations-based models [62, 95, 98] represent the dynamics using a system of coupled differential equations, relying on the complete mixing assumption for the population within a compartment. These models are easier to set up even for a national scale [62].

The second class of models is network-based [27, 35, 39, 57, 59], which are considered in this dissertation. Such models have been found to be more powerful and useful for epidemic spread on large heterogeneous populations, where the complete mixing assumptions of the differential equation models is not reasonable [27, 35, 39, 57, 59]. However, these are harder to set up, simulate, and computationally challenging.

3.2 Prior works on intervention strategies

Vaccination and social distancing are standard intervention strategies used by public health experts to contain epidemics. In this dissertation, we are primarily concerned with vaccinations.

3.2.1 Interventions in differential equation-based models

Much of the work on designing vaccination strategies to control epidemics has been done on differential equation-based models [62, 95, 98]. Optimal vaccination strategies for such models have been analyzed, and compared with the current Centers for Disease Control and Prevention (CDC) policies, for instance, for Swine flu [62]. Even such models tend to be quite complex when the number of compartments becomes very large [95, 98]. Such models can be solved optimally by brute force when they are relatively small (see [62]). However, these don't scale when the model becomes very large, e.g., representing all counties in the United States (US). Greedy strategies have been used in some studies [95, 98], which are relatively easy to implement. More sophisticated gradient descent-based methods have also been designed [9].

3.2.2 Interventions in network-based models

Vaccination in a network-based model corresponds to removing the set of nodes to vaccinate from the network. The EpiControl type problems involve selecting a set of nodes of size at most B (where B is the budget) from the network to vaccinate such that a certain objective function (e.g., expected number of infections) is optimized. Such problems are computationally hard and even obtaining good approximation guarantees is challenging. The previous approaches for designing intervention strategies in these models can be summarized as follows.

Heuristics. As obtaining optimal interventions over network-based models is

computationally challenging, a number of heuristics [20, 25, 65, 101] have been proposed. These methods are based on the idea that the nodes that have high values for certain network properties such as degree, centrality, eigenscore, and pagerank [20, 65, 67, 81]) are ideal candidates for vaccination, so they prioritize them. For instance, the degree-based heuristic involves selecting the top B nodes with the highest degree. These heuristics are extremely simple and can be computed very efficiently even on large networks. Also, these approaches work for any network model. Recently, an influence-based approach of [66, 67] has been parallelized using clever hill climbing techniques. These methods do not directly provide any guarantees for the EpiControl problem.

Optimizing Spectral properties. Spectral radius, denoted by $\rho(G)$ or λ_1, is the largest eigenvalue of the adjacency matrix of the network G. The spectral radius $\rho(G)$ has important implications on the length of the epidemic, where results of the following form are known [32, 63, 76, 97]: if $\rho(G)$ is below a certain threshold value, the disease dies out quickly. Therefore, an important class of intervention design methods focused on reducing the spectral radius of the networks [71, 77–79, 81, 103, 104]. These methods too do not provide bounds for the EpiControl problem. But, such methods can be implemented in polynomial time using eigenvector solvers, or greedy approaches [81].

Approximation algorithms for $p = 1$ case of EpiControl. The EpiControl problem for $p = 1$ and fixed set of sources S is well studied and bicriteria approximation algorithms are known for this problem [28, 41]. Hayrapetyan et al. show that this special case of EpiControl problem is shown to be **NP**-hard via a reduction from the node version of Minimum-Size Bounded-Capacity Cut (MinSBCC) problem.

The edge version of the MinSBCC problem can be summarized as follows: Given a network $G = (V, E)$ with edge capacities c_e, source and sink nodes s and t, budget B. The goal is to find a s-t cut $(P, \overline{P})$, $s \in P$ of capacity at most B

such that the number of nodes on the source side of the cut is minimized. The node version of this problem can be reduced to EpiControl [41].

Hayrapetyan et al. [41] provide a randomized rounding algorithm with a parameter λ for this problem. The variables in the problem are: x_v is an indicator variable to denote which side of the cut $(P, \overline{P})$ node v belongs to; y_e is an indicator variable to denote whether edge e is in cut or not. An LP relaxation of the problem is provided. Then, the idea of this algorithm is very simple:

(i) solve the LP relaxation of the problem to obtain an optimal fractional solution (x^*, y^*) to the problem instance.

(ii) choose $\ell \in [1 - \lambda, 1]$ uniformly at random.

(iii) any node with $x_v^* \geq \ell$ is added to the set P.

This algorithm obtains a $(\frac{1}{\lambda}, \frac{1}{1-\lambda})$ bi-criteria approximate solution for this problem. This result extends to the EpiControl problem for the case with transmission probability $p = 1$. Similar results are shown by Eubank et al. [28].

Interventions for Firefighter problems. Firefighter problems were first introduced in 1995 by Bert Hartnell [33]. The problem can be informally summarized as follows: Given a network $G = (V, E)$, assume that the fire breaks out at a vertex in G at time $t = 0$. Firefighters placed at a node at any time step can defend that node from burning. At each subsequent step, the fire spreads from a burning node to all of its undefended neighbors. Once a node is burning, it will remain so throughout the process. The fire stops when it can no longer spread.

There are many objectives of interest such as (i) saving the maximum number of nodes; (ii) minimizing the expected number of nodes burned (with the fire breaking at a random vertex), etc. Firefighter problem with the objective of minimizing the expected number of nodes burned, where only B firefighters are available, can be viewed as the EpiControl problem on SI model for the case $p = 1$ [3, 29]. Rigorous bounds are known for the number of people infected and saved. However, this has not been much studied for the case where $p < 1$.

Authors in [92] study the problem for $p < 1$ case, but their results are applicable only for the case when G is a tree.

Static interventions. Static interventions in SIR models are known [103, 104]. But these approaches also do not directly bound the expected outbreak size. A special case of this problem is with the work of [6], which considers EpiControl but with the intervention specified at time 0.

Markov Decision Processes (MDP) for interventions. The works based on Markov Decision Processes [3, 16, 29] are able to capture more complex type of interventions. However, these are not very efficient in terms of running time. Linear programming-based techniques are used as subroutines in many of these works, including [3]. As mentioned in the case of saaRound [83] algorithm, linear programming solvers such as Gurobi do not scale to very large networks.

Summary of the novelty of our work on EpiControl problem. None of the above works address the EpiControl for $p < 1$ case directly. Our saaRound algorithm provides bicriteria guarantees on both the budget and the expected number of infections objective for the EpiControl problem. Also, our MwuRound is able to scale to networks with over 100,000 nodes, without losing much on the bounds. The scalable and memory-efficient heuristic based on MwuRound, is able to scale to very large networks with many millions of nodes corresponding to country-level populations. We are able to show that this approach, too, has good guarantees in practice.

Summary of novelty related to IAScontrol. As discussed above, much of the work on designing intervention algorithms focused on the SIR class epidemic models. The MultiPath model is a complex epidemic model with a different structure. None of these works directly provide any guarantees for the IAScontrol problem. Our SpreadBlocking is able to provide a bicriteria approximation for this problem. Further, the performance of this algorithm is near-optimal in practice.

Chapter 4

SAAROUND algorithm for EpiControl Problem

This chapter presents the saaRound [83] algorithm for the EpiControl problem. saaRound algorithm was designed using the sample average approximation (SAA), linear programming, and rounding techniques. In Section 4.1, first, a brief summary of the results in this chapter is presented. In Section 4.2, the intuition behind the algorithm, its description for the case with transmission probability $p < 1$ and a fixed set of source infections, and the analysis of the algorithm are provided. Next, this section provides the extensions of this algorithm to case with source distribution and the two-stage version of EpiControl problem. Further, this section discusses methods to improve the performance of the saaRound algorithm. Finally, Section 4.3 presents the empirical evaluation of the saaRound algorithm.

4.1 Summary of Results

Our results are summarized below:

1. We designed the saaRound algorithm for selecting a set of nodes within

a given budget, to vaccinate at the start of the epidemic. We showed that saaRound gives a $(O(\log nN), 6)$-bicriteria approximate solution for the EpiControl problem, where N is the maximum number of paths from a source node to any node in a set of M sampled subgraph of G. Typically, N is significantly smaller than the number of paths in G, so that in practice, saaRound has a much smaller approximation ratio. saaRound algorithm approach obtains the empirical approximation guarantee of the solution, for any instance of the problem, by comparing it with the LP objective.

2. We showed that saaRound is a good heuristic for the two-stage intervention problem as well, and gives similar guarantees as to the single-stage when the disease transmission subgraphs are trees (e.g., when p is low).

3. We augmented saaRound with a sparsification step, which significantly reduces the size of the LP, and allows scaling to networks, with millions of edges, corresponding to county size population.

4. We evaluated our algorithms on diverse real and random networks. We showed that saaRound has empirical approximation factors very close to 1. These empirical guarantees were significantly better than the worst-case guarantees we have proved rigorously. Further, we showed that the saaRound algorithm outperforms two of the most commonly used baselines for this problem.

5. We examined the network characteristics of nodes in the *intervention set*, as these interventions are near-optimal in practice. These characteristics can help identify "surrogates" for interventions in real-world settings.

4.2 Algorithm

This section presents saaRound algorithm (Algorithm 1) and its analysis. Table 4.1 summarizes the notation for the EpiControl problem.

Notation	Definition
$G = (V, E)$	Contact network
S	Set of sources
p, $p(u, v)$	Transmission probability
$H^{(sir)}$	Stochastic outcome from the SIR process
x_u	Indicator for node v getting vaccinated
$\mathbf{X}$	Set of vaccinated nodes or *intervention set*
$H^{(sir)}(\mathbf{X})$	$H^{(sir)}$ when nodes in $\mathbf{X}$ are vaccinated
$\#\text{infections}(G, S, p, \mathbf{X})$	number of infections in a stochastic outcome when $\mathbf{X}$ is *intervention set*
$\mathbb{E}[\#\text{infections}(G, S, p, \mathbf{X})]$	Expected number of infections when $\mathbf{X}$ is *intervention set*
B	number of vaccines available, called budget
EpiControl	Designing interventions to minimize $\mathbb{E}[\#\text{infections}(G, S, p, \mathbf{X})]$ such that $\mathbf{X} \leq B$
(α, β) approximation	Bicriteria approximation factors

Table 4.1: Summary of notation for the EpiControl problem.

Algorithm 1 describes the steps in saaRound. The algorithm first constructs M sampled outcomes of the SIR process with given transmission probability p. Then, it solves a linear program relaxation of the EpiControl on these M samples.

The variables in the linear program are follows. x_u are indicators for node u getting vaccinated, as defined in Section 2. The variables y_{vj} are indicators for node v getting infected in sampled graph H_j (i.e., there is a path from S to v in H_j on which no nodes are vaccinated). We first describe the intuition behind the algorithm and then analyze its performance.

4.2.1 Intuition behind saaRound

Our algorithm involves five key ideas, which are described below, along with an intuitive description of the steps of the algorithm.

1. **Sampling process.** We first observe that the sampling process in Step 1 of Algorithm 1, which is based on *percolation*, is "equivalent" to the SIR process. The SIR process is a dynamic process in which the state of the network evolves

Algorithm 1 saaRound
Input: $G = (V, E), S, p, B$
Output: $\mathbf{X}$

1: Construct sampled graphs $H_j = (V, E_j)$, for $j = 1, \ldots, M$, by picking each edge $e \in E$ to be in E_j with probability p. Here, $E_j \subseteq E$ denotes the subset of edges picked to be in H_j.
2: Solve the following linear program (LP_{saa})

$$(LP_{saa}) \qquad \min \frac{1}{M} \sum_j \sum_v y_{vj} \tag{4.1}$$

$$\forall j, \forall u \in V: \; y_{uj} \leq 1 - x_u \tag{4.2}$$

$$\forall j, \forall u \in V, \; (w,u) \in E_j: \; y_{uj} \geq y_{wj} - x_u \tag{4.3}$$

$$\forall j, \forall s \in S: \; y_{sj} = 1 \tag{4.4}$$

$$\sum_{u \in V} x_u \leq B \tag{4.5}$$

$$\forall u \; x_u, \; \forall (v,j) \; y_{vj} \in [0,1] \tag{4.6}$$

3: Let x, y be the optimal fractional solution to (LP). We round it to an integral solution X, Y in the following manner

   1. For each (v, j), set $Y_{vj} = y_{vj}$, if $y_{vj} \in \{0, 1\}$. Similarly, for each $u \in V$, set $X_u = x_u$, if $x_u \in \{0, 1\}$.
   2. For each (v, j), round $Y_{vj} = 1$ if $y_{vj} \geq \frac{1}{2}$, otherwise set $Y_{vj} = 0$.
   3. For each u, set $X_u = 1$ with probability $\min\{1, 2x_u \log(4nMN)\}$, where N is the maximum number of paths from S to any node in any sample.
   4. $\mathbf{X} = \{u : X_u = 1\}$ is the set of nodes vaccinated.

4: **return** X

over time. Percolation is an equivalent but a static view of this process, where all edges on which the disease is transmitted are selected in advance. Then, *a node will become infected during the SIR epidemic if and only if there is a path, from a source in S to that node, which consists of only those edges that are sampled in advance.*

2. **Sample average approximation (SAA) technique**: The SAA technique [91] is an approach used to solve stochastic optimization problems. The basic idea is that solving the problem on the samples is enough to get a "good" solution for the stochastic objective. We adapt this technique for the EpiControl problem and show that it suffices to get a solution that minimizes the average number of infections in a set of M sampled outcomes, in order to minimize the expected outbreak size (given by $\mathbb{E}[\#\text{infections}(\cdot)]$) objective, which is an expectation over all the possible outcomes. In our analysis, we show that it suffices that M is bounded by a polynomial in n. We show that using the structure of the SIR model, it suffices to work with M sampled subgraphs H_j for $j \in \{1, \cdots, M\}$, instead of the stochastic outcomes of the SIR process.

3. **Compact integer program**:

   The problem is challenging even if we have to minimize the average number of infections restricted to $H_1, \ldots, H_M$. We start with an integer program (IP) which expresses the following constraints: if a node v is not infected in H_j (which is indicated by $y_{vj} = 0$), then for every path P from a node in S to v in H_j, there must be some node u on the path which has been vaccinated. However, such an integer program would have exponentially many constraints — one for each path. Instead, we design a more compact program (referred to as IP_{saa}), simply based on states of nodes on an edge, as expressed in constraints (4.3), where we adapt the idea presented in [41] to our problem.

4. **Linear relaxation**: We consider a linear relaxation of IP_{saa}, referred to as

LP_{saa}, by replacing the binary constraints by (4.6). LP_{saa} involves minimizing a linear objective over a convex polytope, and so step 2 of Algorithm 1 can be done efficiently to compute the fractional solutions x, y. Also note that since LP_{saa} is optimizing over a larger space (specifically, the convex hull of all the feasible integral solutions), the objective value in (4.1) might be smaller than the integral objective value.

5. **Rounding to an integral solution**: If the solution computed by LP_{saa} is integral, we are done (Step 3(1)). However, in general solution x is fractional, which poses a problem: if we have $x_u \in (0, 1)$, e.g., a fractional value of 0.2, it is not clear how to construct a valid integral solution. In Step 3(2) of saaRound, we pick all the nodes with $y_{vj} \leq 1/2$ (for which $Y_{vj} = 0$), and pick a set of nodes to vaccinate (Step 3(3)), such that every node v with $Y_{vj} = 0$ gets disconnected from S. Step 3(3) achieves this by rounding the fractional solution x, after appropriate scaling. This randomized rounding step ensures that the budgets are not violated by much. This also implies that any node v which gets infected in sample H_j has $y_{vj} \geq 1/2$, so that the average number of infections can be bounded by at most twice the fractional objective value.

4.2.2 Analysis of saaRound algorithm

For a sample H_j computed in Step 1 of saaRound, let
$f(H_j(\mathbf{X})) =< U(0), \ldots, U(\tau), E'_j >$ be defined in the following manner:

(1) E'_j is the subset of E_j when nodes in $\mathbf{X}$ are removed from G (i.e., vaccinated),

(2) $U(0) = S$, and

(3) for $t > 0$, $U(t)$ is the set of nodes at distance t in the subgraph induced by E'_j.

We first observe that the sampling process is "equivalent" to the SIR process.

Observation 6. For any given outcome denoted by $\mathcal{O} =< U(0), \ldots, U(\tau), E'_j >$, we have $\Pr[H^{(sir)}(\mathbf{X}) = \mathcal{O}] = \Pr[f(H_j(\mathbf{X})) = \mathcal{O}]$.

For a vaccination set $\mathbf{X}$, let $Z_j(\mathbf{X})$ be the number of nodes in $H_j - \mathbf{X}$, which are still reachable from S; note that this includes the sources themselves. From Observation 6, it follows that $Z_j(\mathbf{X})$ is equal to the number of infections in the stochastic outcome $H^{(sir)}(\mathbf{X})$ of the SIR process.

Let $Z(\mathbf{X}) = \frac{1}{M}\sum_j Z_j(\mathbf{X})$. Let $\hat{X}_{opt} = \text{argmin}_{X'} Z(X')$ be a solution that achieves the minimum average number of infections in the samples.

Let $X_{opt} = \text{argmin}_{X'}\, \mathbb{E}[\#\text{infections}(\mathbf{X}')]$ be the optimal solution to the EpiControl problem instance. The following lemma shows that the average number of infections achieved by any intervention set $\mathbf{X}$ restricted to the samples $H_1, \ldots, H_M$ is close to the expected number of infections, $\mathbb{E}[\#\text{infections}(\cdot)]$ objective.

Lemma 7. *Let $Z(\cdot)$ be as defined above. If $M \geq 24n^2 \log n$, with probability at least $1 - 1/n$, for every intervention set $\mathbf{X}$, we have,*

$$Z(\mathbf{X}) \in \left[\frac{1}{2}\mathbb{E}[\#\textit{infections}(\mathbf{X})],\ \frac{3}{2}\mathbb{E}[\#\textit{infections}(\mathbf{X})\right].$$

Proof. From Observation 6, we have $E[Z(\mathbf{X})] = E[Z_j(\mathbf{X})] = \mathbb{E}[\#\text{infections}(\mathbf{X})]$ for all j. The $Z_j(\mathbf{X})$ variables are independent, and $\frac{Z_j(\mathbf{X})}{n} \in [0, 1]$. This implies the Chernoff bound (Theorem 5) can be applied to $M\frac{Z(\mathbf{X})}{n} = \sum_j \frac{Z_j(\mathbf{X})}{n}$, so that

$$\Pr\left[\frac{MZ(\mathbf{X})}{n} \notin \left[\frac{1}{2}, \frac{3}{2}\right]\frac{M\mathbb{E}[\#\text{infections}(\mathbf{X})]}{2n}\right] \leq 2exp(-\frac{M}{12n}\mathbb{E}[\#\text{infections}(\mathbf{X})]).$$

We have $\mathbb{E}[\#\text{infections}(\mathbf{X})] \geq 1$, since there is always at least one infection. For $M = 24n^2 \log n$, this probability is at most $2e^{-2n\log n} = \frac{2}{n^n n^n}$. The number of possible intervention sets is the number of possible sets $\mathbf{X} \subseteq V$, which is at most 2^n. Therefore, the probability that there exists an intervention set $\mathbf{X}$ for which

$Z(\mathbf{X}) \notin \left[\frac{1}{2}\mathbb{E}[\#\text{infections}(\mathbf{X})], \frac{3}{2}\mathbb{E}[\#\text{infections}(\mathbf{X})]\right]$ is at most $2^n \cdot \frac{2}{n^n n^n} \leq \frac{1}{n}$ for $n > 1$. □

Recall that IP_{saa} is the integral version of LP_{saa}, obtained by requiring all the variables to be integral, instead of constraints (4.6). We first show that IP_{saa} is valid.

Lemma 8. *For every feasible intervention set* $\mathbf{X}$*, there exists a feasible integral solution* $\bar{x}, \bar{y}$ *to* IP_{saa}*, such that* $\frac{1}{M}\sum_j \sum_v \bar{y}_{v,j} = Z(\{v : \bar{x}_v = 1\})$*. If* $\bar{x}, \bar{y}$ *is an optimal solution to* IP_{saa}*,* $Z(\hat{X}_{opt}) = \frac{1}{M}\sum_j \sum_v \bar{y}_{v,j}$

Proof. First, consider a feasible intervention set $\mathbf{X}$. We define $\bar{x}_v = 1$ for all $v \in \mathbf{X}$. We define $\bar{y}$ in the following manner: Let $f(H_j(\mathbf{X})) =< U_j(0), \ldots, U_j(\tau_j), E'_j >$, as defined earlier; we have $Z_j(\mathbf{X}) = \sum_t |U(t)|$. We define $y_{vj} = 1$ if $v \in \cup_t U_j(t)$.

Now, we show that $\bar{x}, \bar{y}$, defined in the above manner, is a feasible solution to IP_{saa}. For any j, consider a node $u \in U_j(t)$ for some t. Then, there exists a path $P = u_0, u_1, \ldots, u_t = u$ with $u_i \in U_j(i)$ for $i \leq t$. By construction, for each node u_i, we have $y_{u_i j} = 1 \geq y_{wj} - x_{u_i}$ for every neighbor w of u_i, which implies the constraint (4.3) is satisfied for u, and each of its neighbor w. Let $U = \cup_{t=0}^{\tau_j} U_j(t)$. Consider a node $u \notin U$. If u has a neighbor $w \in U$, it must be the case that $u \in \mathbf{X}$, else node u would be infected at time $\tau_j + 1$, and would have been in a set $U(\tau_j + 1)$. This implies, $x_u = 1$, and the constraint (4.3) holds for node u and any neighbor w. If u has no neighbor $w \in U$, then $y_{wj} = 0$, and so the constraint (4.3) holds for u, w.

The converse follows similarly. We need the following additional property: if $y_{uj} = 1$, there is a path P from S with $y_{u_i j} = 1$ for all nodes $u_i \in P$; this holds due to the minimization objective. □

Lemma 9. *For any sampled graph* H_j*, and any node* $v \in V$ *with* $y_{vj} < \frac{1}{2}$*, we have,*

$$\Pr[v \text{ is reachable from } S \text{ in } H_j[V - \mathbf{X}]] < \frac{1}{4nM},$$

where $H_j[V - \mathbf{X}]$ is the graph induced by removing the nodes in $\mathbf{X}$ from H_j.

Proof. Let $\mathcal{P}_{vj} = \{P_1, \ldots, P_L\}$ be the set of paths to node v in the sampled graph H_j. For a path P, define $S(P) = \{u : u \in P\}$ to be set of nodes on the path P. Node $v \in V$ is reachable from S in $H_j[V - \mathbf{X}]$ if and only if there exists some path $P \in \mathcal{P}_{vj}$ such that none of the nodes in $S(P)$ are vaccinated (i.e., $X_u = 0$, $\forall u \in S(P)$). If there exists $u \in S(P)$ with $2x_u \log(4nMN) \geq 1$, the rounding ensures that $X_u = 1$; therefore, we only consider the case $2x_u \log(4nMN) \leq 1$. Our rounding ensures that we have $\Pr[X_u = 1] \geq 2x_u \log(4nMN)$, so that $\Pr[\sum_{u \in S(P)} X_u = 0]$ is upper bounded by

$$\prod_{u \in S(P)} \big(1 - 2x_u \log(4nMN)\big) \leq e^{-\sum_{u \in S(P)} 2x_u \log(4nMN)} \leq e^{-\log(4nMN)} = \frac{1}{4nMN} ,$$

since $\sum_{u \in S(P)} x_u \geq 1 - y_{vj} \geq 1/2$.

Equivalently, the probability that no node from $S(P)$ is picked is at most $\frac{1}{4nMN}$; here we consider that a node is picked from $S(P)$, if $X_u = 1$ for some $u \in S(P)$. By a union bound, the probability that there exists a path $P \in \mathcal{P}_{vj}$ such that no node from $S(P)$ is picked is at most $\frac{L}{4nMN} \leq \frac{1}{4nM}$ (since, $L \leq N$). Hence, the lemma follows. □

Lemma 10. *With probability at least $1 - 1/n$, we have $|\mathbf{X}| \leq 12 \log(4nMN)B$.*

Proof. Let X be the rounded solution returned by saaRound algorithm which corresponds to the intervention set $\mathbf{X} = \{u : X_u = 1\}$. Then, the expected number of nodes picked for vaccination by saaRound is given by

$$\mu = E\left[\sum_u X_u\right] \leq \sum_u 2x_u \log(4nMN) \leq 2 \log(4nMN)B$$

The first inequality is by linearity of expectation and the second inequality follows from the constraint (5) of LP_{saa}. The X_u's are all rounded independently,

therefore, we have

$$\begin{aligned}\Pr\left[\sum_u X_u > 12\log(4nMN)B\right] &\leq \Pr\left[\sum_u X_u \geq 6\mu\right] \\ &\leq exp(-6\log(4nMN)B) \\ &\leq \frac{1}{n}.\end{aligned}$$

The first inequality follows from the bound on μ. The second inequality follows from the Chernoff bound (Theorem 5), as $6\mu \geq 2e\mu$. Finally, the last inequality follows, since $6\log(4nMN)B \geq \log n$. □

Theorem 11. *Let $M \geq 24n^2 \log n$. Let $\mathbf{X}$ denote the vaccination set computed by the* saaRound *algorithm. Then, with probability at least $1/2$, we have*

$$\mathbb{E}[\#infections(\mathbf{X})] \leq 6\mathbb{E}[\#infections(X_{opt})],$$

and $|\mathbf{X}| \leq 12\log(4nMN)B$.

Proof. Let $\hat{X}_{opt}$ be as defined above. By Lemma 9, for any v, j, if $y_{vj} \leq 1/2$, the probability that node v is reachable from S is at most $\frac{1}{4nM}$. By a union bound, the probability that this holds for at least one vertex $v \in V$ (for a fixed j) is at most $\frac{1}{4M}$. This implies that with probability at least $1 - \frac{1}{4M}$,

$$Z_j(\mathbf{X}) \leq |\{v : y_{vj} \geq 1/2\}| \leq \sum_{v:y_{vj}\geq 1/2} 2y_{vj} \leq \sum_v 2y_{vj}$$

By a union bound, with probability at least $1 - \frac{M}{4M} = 1 - \frac{1}{4}$, we have $Z_j(\mathbf{X}) \leq 2\sum_v y_{vj}$, for all j. By definition of $\hat{X}_{opt}$, we have $\frac{1}{M}\sum_j Z_j(\mathbf{X}) \leq \frac{1}{M}\sum_{v,j} 2y_{vj} \leq 2Z(\hat{X}_{opt})$, since the LP solution is also a lower bound on $Z(\hat{X}_{opt})$. By Lemma 10, the condition $|\mathbf{X}| \leq 12\log(4nMN)B$ holds, in addition to $Z(\mathbf{X}) \leq 2Z(\hat{X}_{opt}) \leq 2Z(X_{opt})$, with probability at least $1-\frac{1}{4}-\frac{1}{n}$, since $Z(\hat{X}_{opt}) \leq Z(X_{opt})$, by definition of $\hat{X}_{opt}$.

By Lemma 7, with probability at least $1 - \frac{1}{n}$, we have,

$$Z(X_{opt}) \leq \frac{3}{2}\mathbb{E}[\#\text{infections}(X_{opt})] \tag{4.7}$$

and $\frac{1}{2}\mathbb{E}[\#\text{infections}(\mathbf{X})] \leq Z(\mathbf{X})$. This gives us

$$\mathbb{E}[\#\text{infections}(\mathbf{X})] \leq 2Z(\mathbf{X}) \leq 4Z(X_{opt}) \leq 6\mathbb{E}[\#\text{infections}(X_{opt})]$$

Therefore, all the conditions of the theorem hold with probability
$\geq 1 - \frac{1}{4} - \frac{2}{n} \geq \frac{1}{2}$. □

4.2.3 Extension to the case with source distribution

We assume s_v is the probability that v is initially infected; $\mathbf{s}$ denotes the initial infection vector. In this case, we assume sources can also be considered for vaccination unlike the case with a fixed set of sources.

First, we present the construction of sampled graphs for this case. Then, we will present the LP relaxation for the problem on samples. The rounding procedure remains unchanged.

Sampling process. Construct a sampled graph $H_j = (V_j, E_j)$, for $j = 1, \ldots, M$, by picking each edge $e \in E$ to be in E'_j with probability p. Also pick a set of sources $\texttt{src}(H_j)$ (denotes the source set in sample H_j) by sampling from $\mathbf{s}$. Then, V_j denotes all nodes connected to $\texttt{src}(H_j)$ in H_j. The edge set E_j denotes edges in E'_j whose both endpoints are in V_j.

Solve the following modified linear program (LP_{saa})

$$\min \frac{1}{M} \sum_{j} \sum_{v} y_{vj} \tag{4.8}$$

$$\forall j, \forall u \in V: \; y_{uj} \leq 1 - x_u \tag{4.9}$$

$$\forall j, \forall u \in V, \; (w,u) \in E_j: \; y_{uj} \geq y_{wj} - x_u \tag{4.10}$$

$$\forall j, \forall s \in src(H_j): \; y_{sj} = 1 - x_s \tag{4.11}$$

$$\sum_{u \in V} x_u \leq B \tag{4.12}$$

$$\text{All variables} \in [0,1] \tag{4.13}$$

Notice that the main change is that the constraint (4.11) is added. This constraint makes sure that a source node is uninfected if it is vaccinated.

4.2.4 Extension to the multi-stage versions

In this section, we present extension of saaRound for the multi-stage versions of the EpiControl problem. The problem statement for the two-stage version (referred to as 2sEpiControl problem) is as follows.

Definition 12. 2sEpiControl problem.

Instance. Given a contact network $G = (V, E)$, and an initially infected set of nodes S, budgets B_0, B_T for interventions at time $t = 0$ and $t = T$ respectively.

Goal. To find subsets of nodes $\mathbf{X}_0, \mathbf{X}_T \subseteq V$ to intervene at $t = 0$ and $t = T$ respectively, such that $\mathbb{E}[\#\text{infections}(\mathbf{X}_0, \mathbf{X}_T)]$ is minimized, and $|\mathbf{X}_0| \leq B_0$, $|\mathbf{X}_T| \leq B_T$.

Note. The EpiControl version (discussed in previous sections) where all the interventions are performed at the beginning is also referred as 1sEpiControl explicitly when the context is unclear.

Modifications to saaRound for 2sEpiControl. The changes to be made to LP_{saa} to adapt it for 2sEpiControl (our approach can be similarly extended

to multiple stages) are as follows: Let x_{u0} be an indicator variable whether node u is vaccinated at time $t = 0$. Similarly, let x_{uT} be an indicator variable whether node u is vaccinated at time T. We have B_0, B_T as inputs. Let $V_{j,t}$ denote the set of all nodes at level t in the BFS tree in H_j which has the nodes in S at level 0; let $V_{j,\geq t} = \cup_{t' \geq t} V_{j,t}$ denote the set of all nodes at level t or more.

Constraint (4.2) is modified in the following manner: for all nodes u in the set $V_{j,\geq T} - S$ in each sample H_j, we have

$$\forall j, u \in V_{j,\geq T} - S, \forall t : \; y_{uj} \leq 1 - x_{ut}.$$

The Constraint (4.3) is changed to

$$\forall j, \forall u \in V, \; (w,u) \in E_j : \; y_{uj} \geq y_{wj} - \sum_{t: u \in V_{j,\geq t}} x_{ut}.$$

We add the constraint for each $t = T$,

$$\sum_u x_{uT} \leq B_T.$$

We refer to this modified LP_{saa} as LP^e_{saa}. We use the same rounding procedure for the x and y variables as in SAAROUND. The algorithm returns two subsets of nodes $\mathbf{X}_0 = \{u : X_{u0} = 1\}$ and $\mathbf{X}_T = \{u : X_{uT} = 1\}$ as the solution.

Analysis: If the sampled subgraphs are trees (which is typical for low transmission probability), LP^e_{saa} is valid, and we can show the same guarantees as Theorem 11. In general, however, LP^e_{saa} may not be valid, and the solution might not have these guarantees, due to the following reason: suppose there is a node u which is at level $< T$ in a sampled subgraph H_j before the first stage of intervention is done at time 0. After a set $\mathbf{X}_0$ is picked (and the nodes in $\mathbf{X}_0$ are removed from the graph), the distance of u from S might increase, and it could be vaccinated at time T. However, our algorithm will not pick such nodes, and

thus optimizes over a smaller decision space.

4.2.5 Improving performance and speeding up saaRound

Improved approximation factor. The worst case approximation is most impacted by the scaling we do in step 4(3) of saaRound, which is needed for the application of the Chernoff bound in Lemma 10. However, as we discuss later, we find that LP_{saa} computes near-integral solutions, in which many variables are integral. Step 4(1) handles integral variables separately. We also modify Step 4(3) by using a smaller scaling factor, depending on the fractional value.

Better scaling. The main bottleneck in saaRound is the solution of LP_{saa}, which has: nM variables of the form y_{uj}, $n|\mathcal{T}|$ variables of the form x_{ut}, and $\sum_j |E(H_j)|$ constraints (4.3). The worst-case dependence of the running time of LP solvers is super-quadratic in these parameters (though we find the Gurobi solver [37] scales very well in practice, as we discuss later). In order to improve the scaling of saaRound to larger instances, we use the following methods.

- Reduced number of samples: The rigorous bound on the number of samples needed in the worst-case comes from Lemma 7, as a result of the Chernoff bound. In practice, we find that there is concentration even with $O(\sqrt{n})$ samples, and so we use fewer samples in our experiments. This can be estimated in a statistically rigorous manner by picking the smallest number of samples such that the variance in infections (i.e., number of reachable nodes from sources S) is within a factor δ.
- Reducing the number of variables: We define *vulnerability* of a node u, denoted by y_u, as the probability that it gets infected when no interventions are done. This can be estimated as the fraction of samples H_j in which u is reachable from S, i.e., $y_u = \frac{1}{M}|\{j : u \text{ is reachable from } S \text{ in } H_j\}$. For a parameter γ, we restrict the interventions to nodes in $V_\gamma = \{v : y_v > 1 - \gamma\}$; in other words, we can set $x_{vt} = 0$ for nodes with vulnerability at most γ. The

intuition is that such nodes are likely to have low x_{vt} values in LP_{saa}, and so it is safe to remove them and reduce the size of the LP. This is borne out from our experiments.

4.3 Experiments

We addressed the following questions in our experiments:

1. **Approximation Guarantees**: What are the approximation factors of SAAROUND in practice? How does it compare with the standard baselines?
2. **Scaling**: How well does SAAROUND scale to large networks? How effective are the techniques for choosing the number of samples and pruning?
3. **Effect of multiple stages**: How does the effectiveness of the solution computed by SAAROUND algorithm vary with the number of stages of interventions and the budget allotted to each stage?
4. **Characteristics of the solutions**: What kinds of nodes are picked in the solutions at each stage? What are the characteristics of nodes in the near-optimal *intervention sets*?

4.3.1 Dataset and Methods

Datasets. We experiment with three different classes of networks (a total of eight), in order to fully explore the effect of network structure on the results.

We consider two random network models, namely the small world [53], and the preferential attachment [7]. The parameters used in the generation of the random networks are as follows: We use Networkx tool [38] to generate the three random graphs PA1, PA2, and SW with the following parameters. Further details of these parameters can be found in Networkx.

1. Preferential1 (PA1): $barabasi\_albert\_graph(n = 1000, m = 2, seed = None)$

2. Preferential (PA2): $barabasi\_albert\_graph(n = 100000, m = 2, seed = None)$

3. Small World (SW): $navigable\_small\_world\_graph(n = 50, p = 1, q = 5, r = 2, dim = 2, seed = None)$

We study the results on the CA-GrQc and CA-HepTh collaboration networks [56] since it is a type of social network. We also consider synthetic agent based populations for Montgomery County, VA, and Portland, OR, constructed by the first-principles approach by [8, 27]. This has been used in several public health studies, e.g., [88]. This network has a rich set of demographic attributes for each node, e.g., age, gender, and income. The details of the networks are summarized in Table 4.2.

Dataset	**Nodes**	**Edges**
Preferential1 (PA1)	1000	1996
Small World (SW)	2500	14833
BTER	4756	35272
CA-GrQc	5242	14496
CA-HepTh	9877	25998
Montgomery	75457	648667
Portland	1409197	8307767

Table 4.2: saaRound algorithm: Description of datasets

Choosing parameters. There is a large space of model parameters over which the analysis could be done. We choose a subset of them as described here. We choose the source distribution $\mathbf{s}$ for seed infections such that the expected number of sources is 10. We assume uniform distribution for $\mathbf{s}$. Then, the probability s_u that a node u in a network is a source is given by $s_u = \frac{10}{n}$, where n is the number of nodes in the network.

Following standard practice in public health, e.g., [39], we choose three values for the transmission probability p based on the expected number of infections, referred to as the "attack rate". Attack rate of a disease is the percentage of population that gets infection. We choose a probability p_{low} if the attack rate is

$< 10\%$ (low), p_{med} if the attack rate is in $[10\%, 20\%]$ (medium), and p_{high} if the attack rate is $> 20\%$. The specific probability values depend on the networks and their structure. Figure 4.1 presents the effect of transmission probability p on the attack rate for different networks. This analysis is useful in identifying the parameter p value range for a particular attack rate.

Methods. We focus on one stage (1SEPICONTROL) or two-stage (2SEPICONTROL) versions of EPICONTROL in our experiments. We use SAAROUND algorithm to find the set $\mathbf{X}$ of interventions. The SAAROUND algorithm uses the Gurobi solver [37] to solve the LP relaxation for the problem instance.

For 1SEPICONTROL problem, we consider the following baselines, which select B nodes based on two different criteria:

- **Top-B degree.** This heuristics picks top B nodes with the highest degree as the nodes for intervention. The intuition is that a node (i.e., person) with many edges (i.e., connections) is likely a better candidate for intervention in order to decrease the total number of infections. This approach is very popular and is considered in a number of papers [7, 82].

- **Top-B EVC.** This heuristic selects the nodes with the highest eigenvector centrality score for intervention. Eigenvector centrality score measures the influence of a node in the network: a high score indicates that the node is connected to many nodes who themselves have high scores.

The top-B EVC does not give insights on the performance of the spectral approaches [77–79, 93, 98, 103, 104]. A more detailed comparison of our approach with the spectral methods is an important future direction. We also propose a new approach called vulnerability based on observations from our experimental results. This is described below.

Vulnerability. We compute the vulnerability of each node u, denoted by y_u, which is the probability that this node gets infected, and select top B nodes with

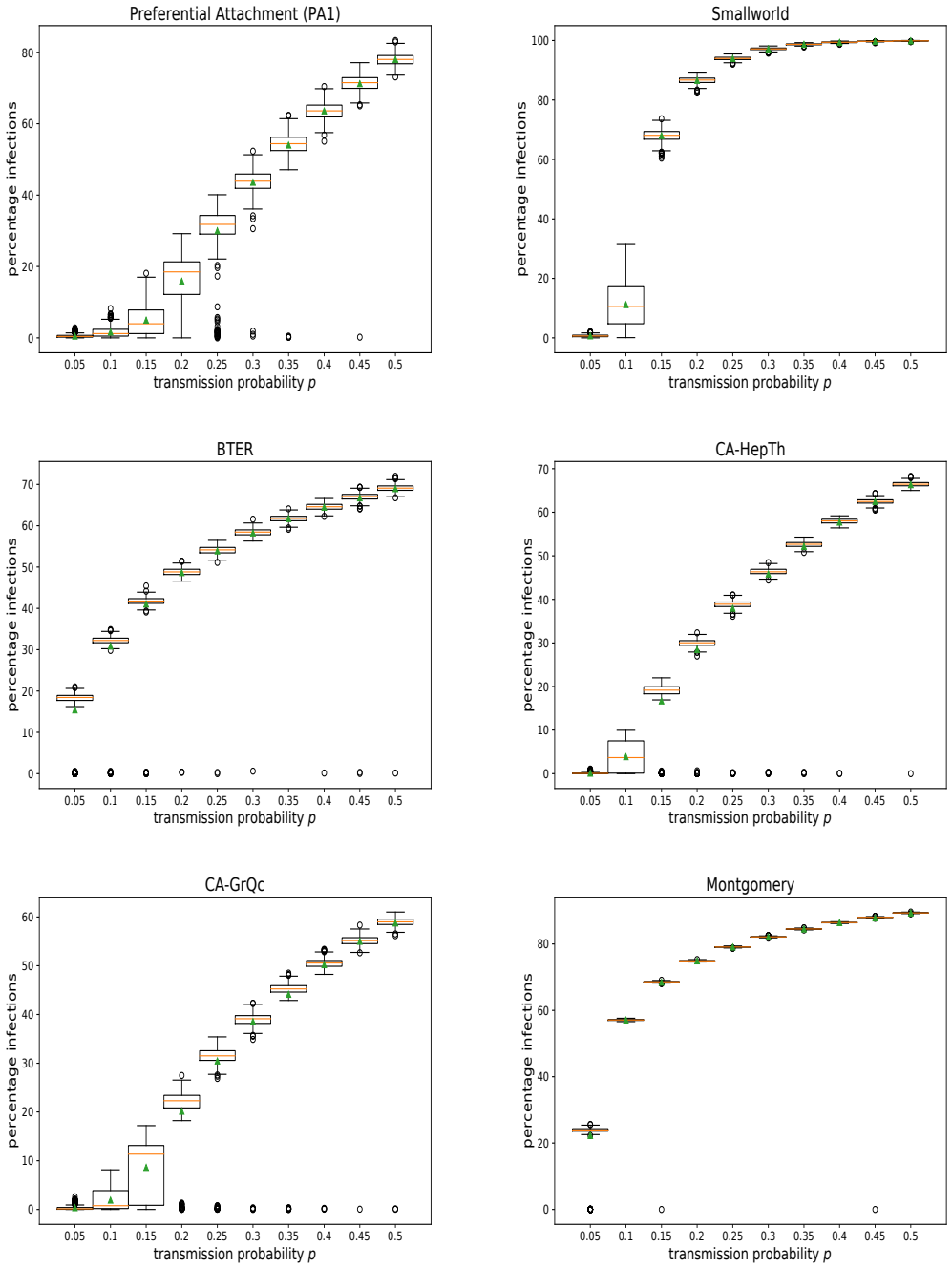


Figure 4.1: The effect of transmission probability p on the attack rate for different networks.

the highest vulnerability.

No prior results are known for 2sEpiControl problem. Therefore, we adapt the above baselines and pick B_0 and B_T nodes in the order of the above scores.

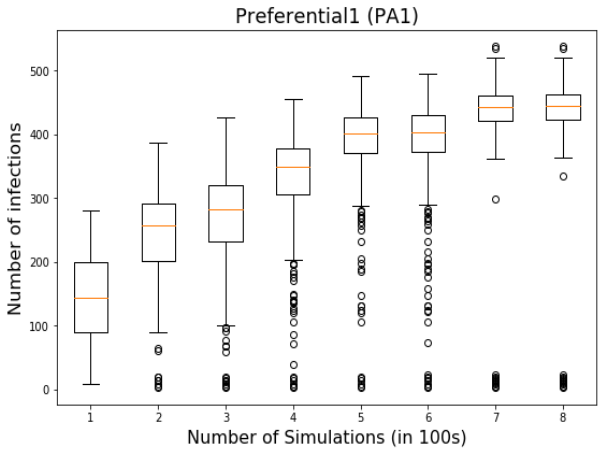


Figure 4.2: Number of simulations needed for low attack rate.

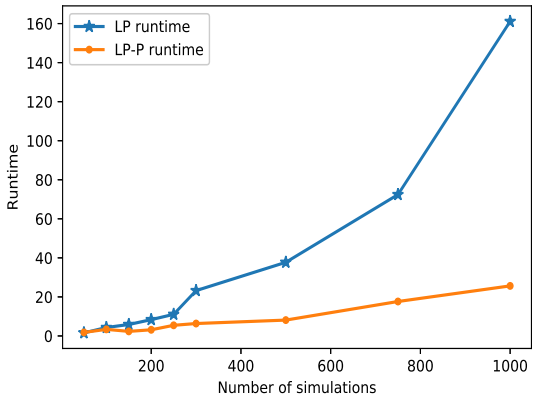


Figure 4.3: Comparison of runtimes (in seconds) of linear program, for an instance, with (LP-P) and without pruning (LP).

4.3.2 Scaling

We find that SAAROUND easily scales well to all the networks considered except Montgomery and Portland. Also, we note that the two strategies for speeding up have a significant impact on the scaling.

- **Number of samples needed**: We find the number of samples sufficient to get reasonable variance, as shown in Figure 4.2 to be less than the worst-case bound of $\Theta(n \log n)$ from Lemma 7. As the number of samples increases, the expected number of infections over those samples converges to (or very close to) the expected value. In practice, we observed that the number of samples needed for this convergence in many cases is $O(\sqrt{n})$ for moderate attack rates

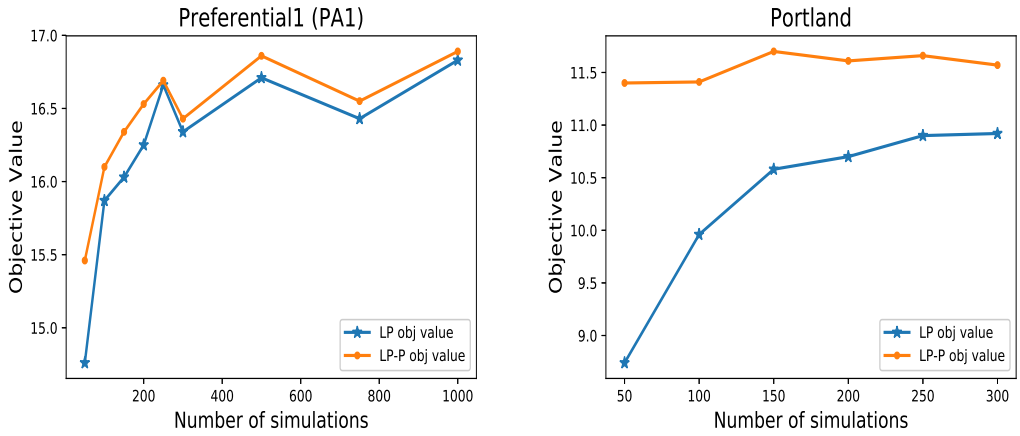


Figure 4.4: Comparison of objective values of linear program, for an instance, with (LP-P) and without pruning (LP).

(10%-20%). The number of samples needed for convergence is lower when the transmission probability p belongs to medium or above attack rates. We note that these are typically the regimes of most concern to public health agencies.

- **Impact of pruning**: The pruning of low vulnerability nodes has a very significant impact on the running time, as shown in Figure 4.3, which shows the running time with and without pruning. When the number of samples used is low, the difference is negligible, but when the number of samples increases to the range needed for low variance, we find the difference in running times is in several orders of magnitude. The objective value differs by less than 5% with and without pruning for PA1 network. Similar trend is observed for Portland network. This can be seen in Figure 4.4. This implies that our scaling strategies give good solutions on large networks.

4.3.3 Performance guarantees and comparison to baselines

Comparison to baselines

Figures 4.5 shows the performance of saaRound in comparison to the baselines and the vulnerability method. The X-axis of these plots represents the budget

B as a percentage of the population (network size n). The Y-axis of these plots represents the average % of the population infected. Each curve corresponds to a baseline. Along with the baselines top-B degree, top-B EVC, and Vulnerability, we also consider the "LP Obj" which is the LP relaxation optimal objective value. This value gives a lower bound on the optimum value of the EpiControl instance on the M samples. Further, we also compare with the "No Action" baseline, which gives the average number of infections with no interventions. The "No Action" also shows the attack rate resulting from p chosen. The attack rate considered for all networks, except Montgomery and PA2, is about 15%. For these two networks, it is about 5%. The probabilities are set accordingly to achieve these attack rates. Every approach uses exactly the same budget as the rounded solution to saaRound does. This is done for the purpose of a fair comparison of performance guarantees.

We observe that saaRound significantly outperforms all the approaches. For social contact networks, which are relatively dense, the objective value from the top-B-EVC and top-B-degree baselines are over seven and three times that from saaRound, respectively, over the entire budget range. In networks such as collaboration (CA-GrQc and CA-HepTh) vulnerability shows performance similar to that of top-B degree baseline. Figures 4.5 shows that for most of the networks, the objective value of LP optimal solution almost coincides with that of saaRound.

Approximation Ratio

As we provide bi-criteria approximation guarantees in the theoretical results, in this section, we evaluate the empirical performance of our algorithm and show the guarantees saaRound achieves in practice.

The ratio of the average number of infections resulting from the interventions set $\mathbf{X}$ computed by the saaRound algorithm to the optimum average number

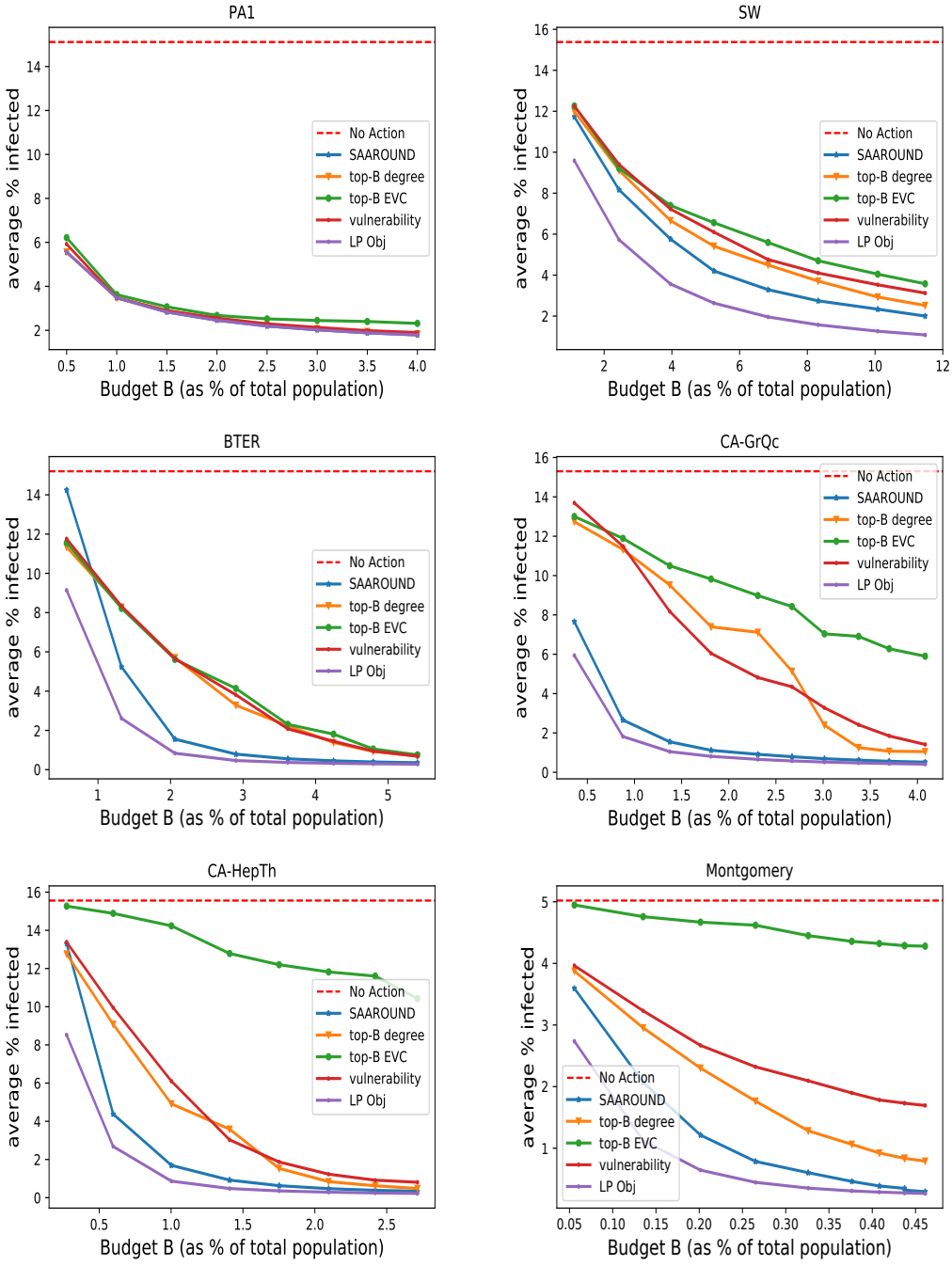


Figure 4.5: Comparison of SAAROUND with baselines top-B degree, top-EVC, and vulnerability. LP Obj corresponds to the lower bound on the optimal obtained by solving the linear programming relaxation. The dashed red line corresponds to the average % infected for the "No Action" (no interventions are performed) scenario.

of infections is referred to as the objective approximation ratio. This optimum can be obtained by solving an Integer Linear Program (ILP) for an instance of the problem on samples. However, the ILP is very slow even on moderate-sized networks, so we use LP Obj, which is the optimum value of the LP relaxation for this problem. Then, we use this in the denominator to compute the objective approximation ratio for the instance. We observe that the approximation ratio with respect to the objective value is close to 1 in most cases, and is at most 2 in all our experiments. This is shown in Figure 4.6 for various networks. Note that the actual optimum can be greater than the LP Obj, so this indicates near-optimal performance.

The ratio of number of interventions in the solution $\mathbf{X}$ returned by saaRound to the given budget B is called *budget violation* or *budget approximation ratio*. Figure 4.7 shows that the budget approximation ratio on various networks. This ratio is very close to 1 for all the networks considered and has a value at most 1.75. This is much better than the theoretical bound we show for this criterion (logarithmic in the number of paths). This shows that saaRound performs much better in practice than the theoretical results we were able to obtain.

4.3.4 Impact of the interventions on the variance of the number of infections in the samples

In this experiment, we remove the intervention set $\mathbf{X}$ computed by saaRound from the network. Then, we re-compute the sampled graphs for the same parameter settings, on this residual network, to get the average number of infections.

Figure 4.8 shows the effect of interventions by saaRound on the variance of the number of infections in the sampled graphs. The left-most box plot in each plot corresponds to the "No Action" scenario, that is no node is removed from the network before generating the simulations (or budget $B = 0$ in this case).

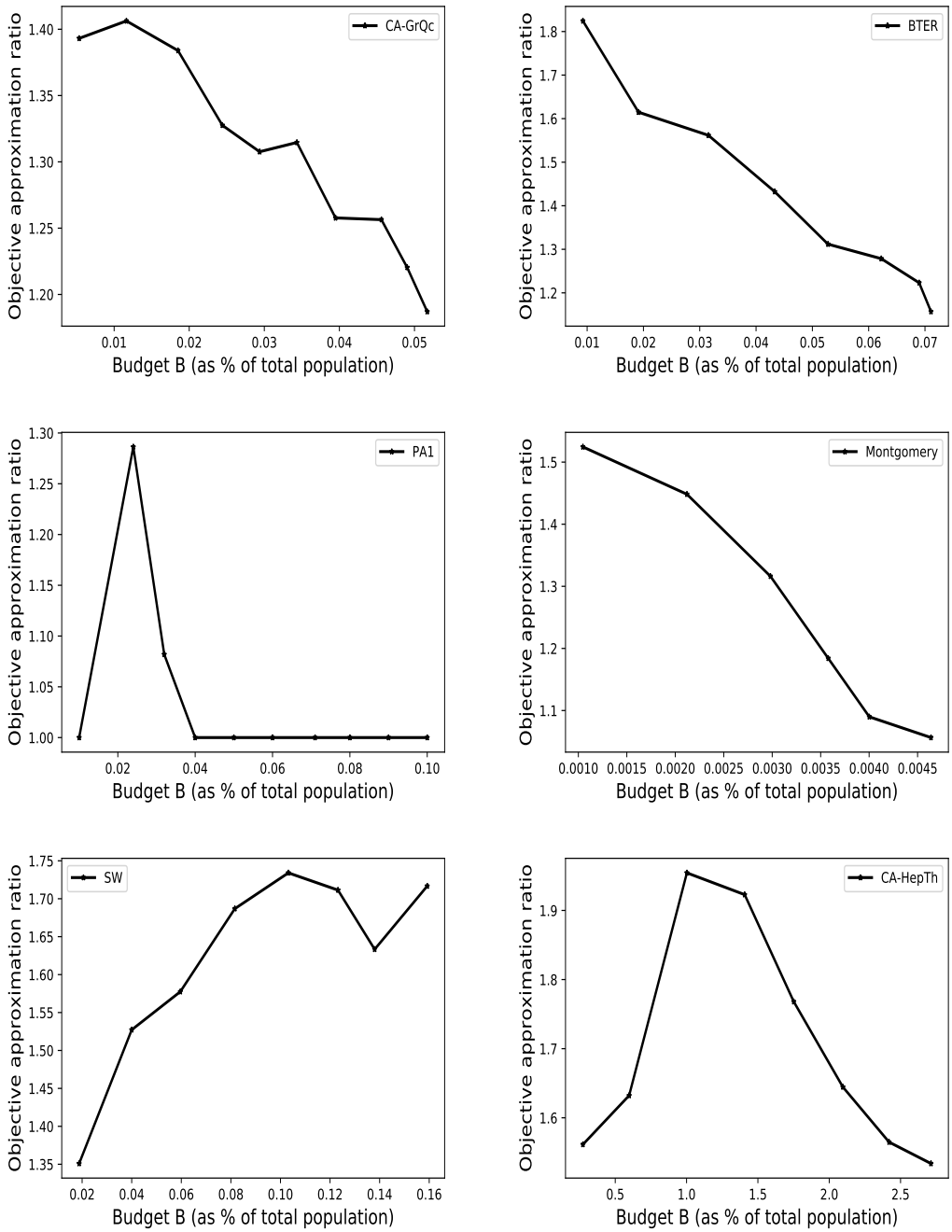


Figure 4.6: Empirical objective approximation ratio of saaRound.

The rest of the box plots correspond to different budgets as a percentage of the population.

The median (red line in the box plot) clearly falls down as the budget (no. of nodes removed) is increased as expected. The variance in the number of infections is captured by the length of the box plot for a particular budget. In

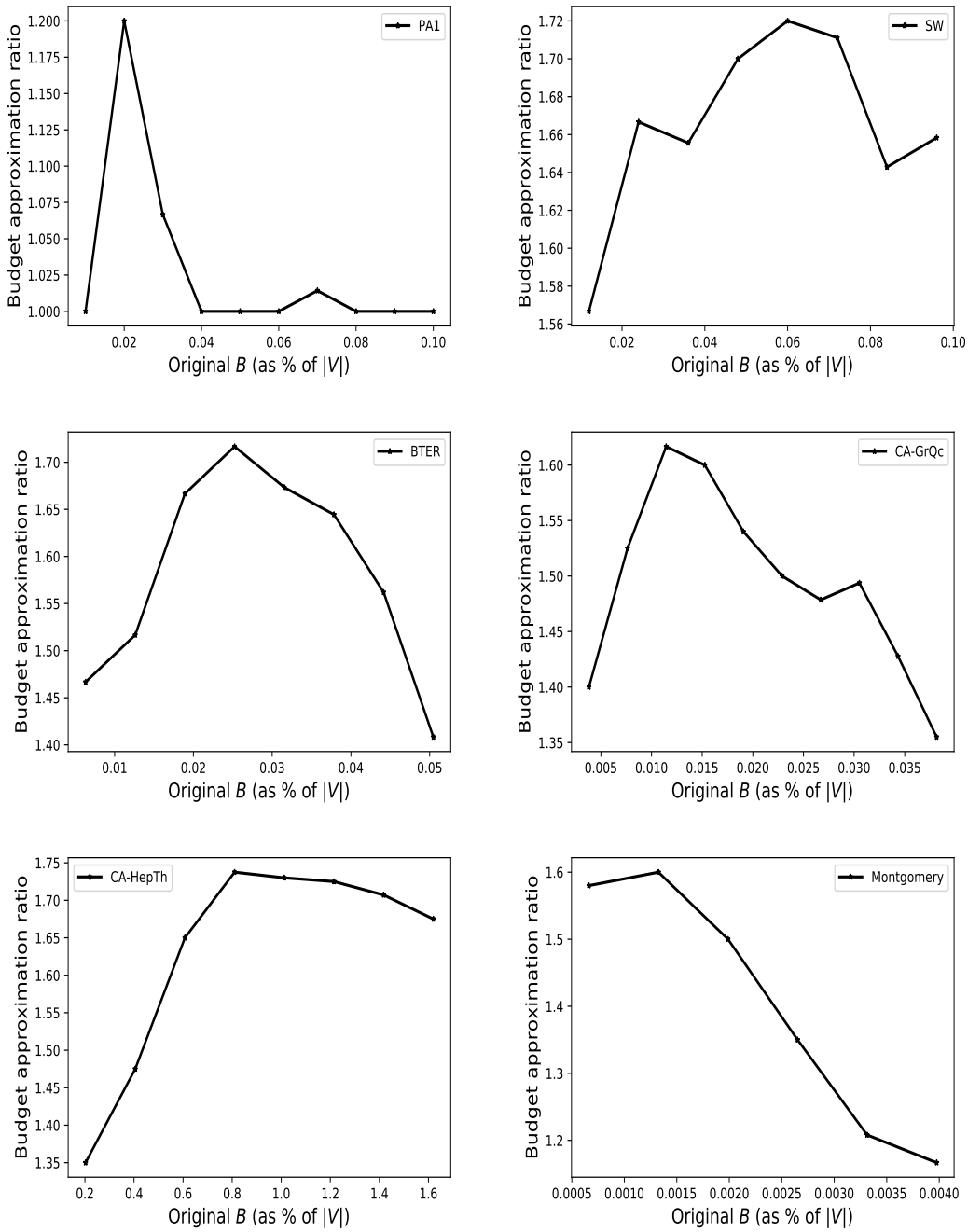


Figure 4.7: Empirical budget approximation ratio of saaRound.

most networks, the variance decreases as the budget is increased. But in some cases, e.g., the collaboration networks, the variance increases for small budgets but decreases sharply as the budget is further increased beyond a threshold. The variance shows that the interventions computed for the expected infections objective, although reducing the average number of infections over the M outcomes,

may not be an ideal solution for some of these outcomes. That is the reason, we see the high variance for smaller budgets in certain networks.

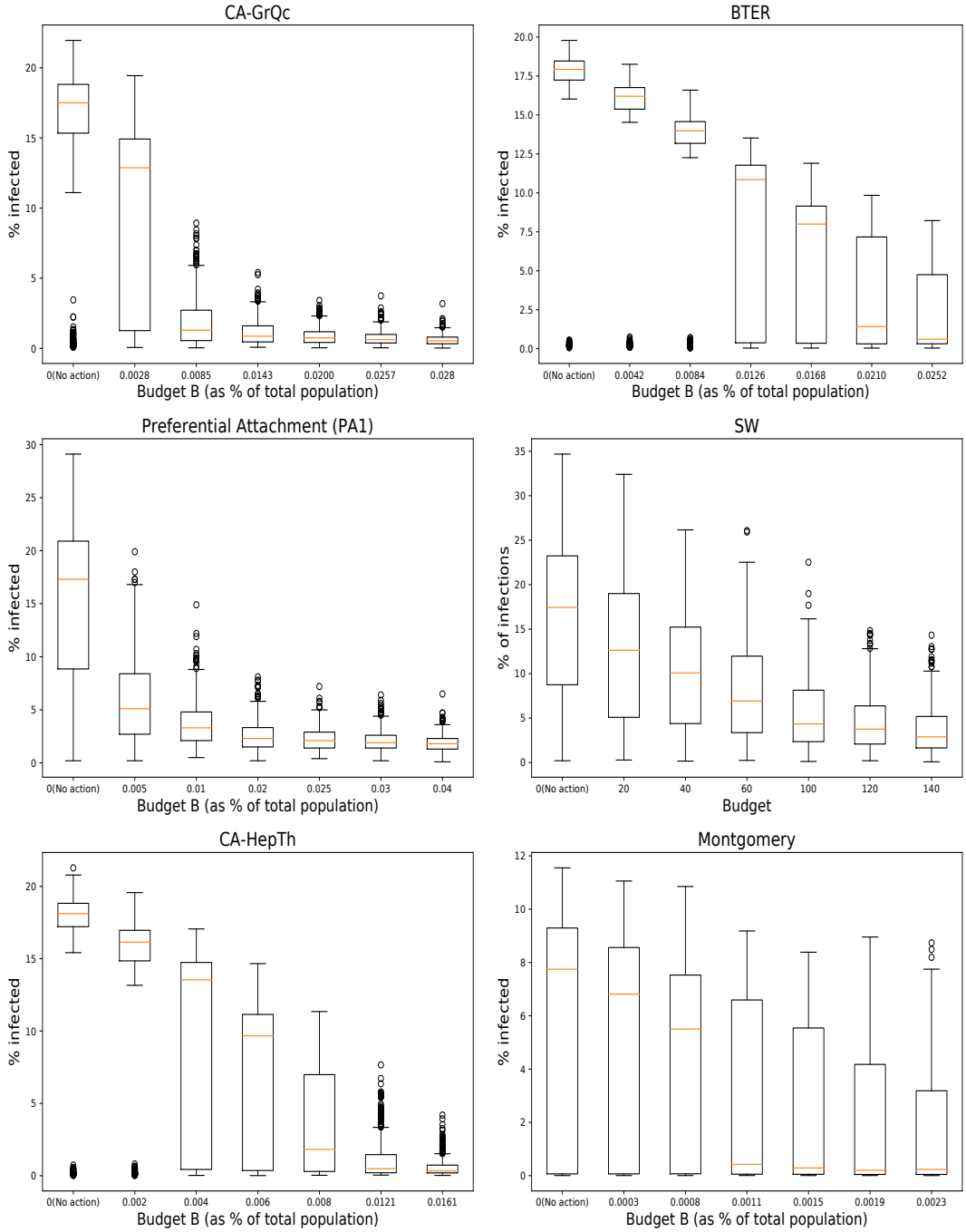


Figure 4.8: Impact of varying budget B on the percentage of infections resulting from the *intervention set* obtain by saaRound.

4.3.5 Characteristics of Near-optimal interventions

As shown in previous sections, the solutions obtained by SAAROUND algorithm tend to be near-optimal in practice. This prompts a question on what network properties the nodes in these solutions possess. Since, targeted interventions (i.e., immunizing particular individuals) are not practical, identifying the network properties of the nodes in near-optimal solutions can help in designing "surrogates" for interventions in real-world settings. We consider two network properties of a node: (i) degree, and (ii) clustering coefficient. The clustering coefficient of a node measures how well-connected its neighborhood is. In other words, this measures how close the neighbors of this node are in forming a clique.

We notice that the nodes picked for interventions by SAAROUND tend to be picked based on a metric that is a combination of their degree and clustering coefficient. This is shown in Figure 4.9. As the budget increases, nodes with less clustering coefficient or degree are also added into the *intervention set*.

Another interesting question is to study the impact of transmission probability p on the network properties of nodes in a near-optimal solution. We fix the budget in this experiment. For smaller values of p, most nodes in the solution have higher degree and clustering coefficient values. But, we notice that, as p value increases, the nodes with a relatively low degree are picked for intervention, whereas for small p values, most of the nodes picked for intervention have a high degree. Particularly, we notice a shift in the average degree of the nodes picked for intervention as p is varied. This is shown in Figure 4.10.

4.3.6 Two stage intervention

In this experiment, we consider the two-stage version (2SEPICONTROL problem) of EPICONTROL. Figure 4.11 shows impact of time step T at which the interventions are performed on the objective value $\mathbb{E}[\#\text{infections}(\mathbf{X})]$. The first stage of

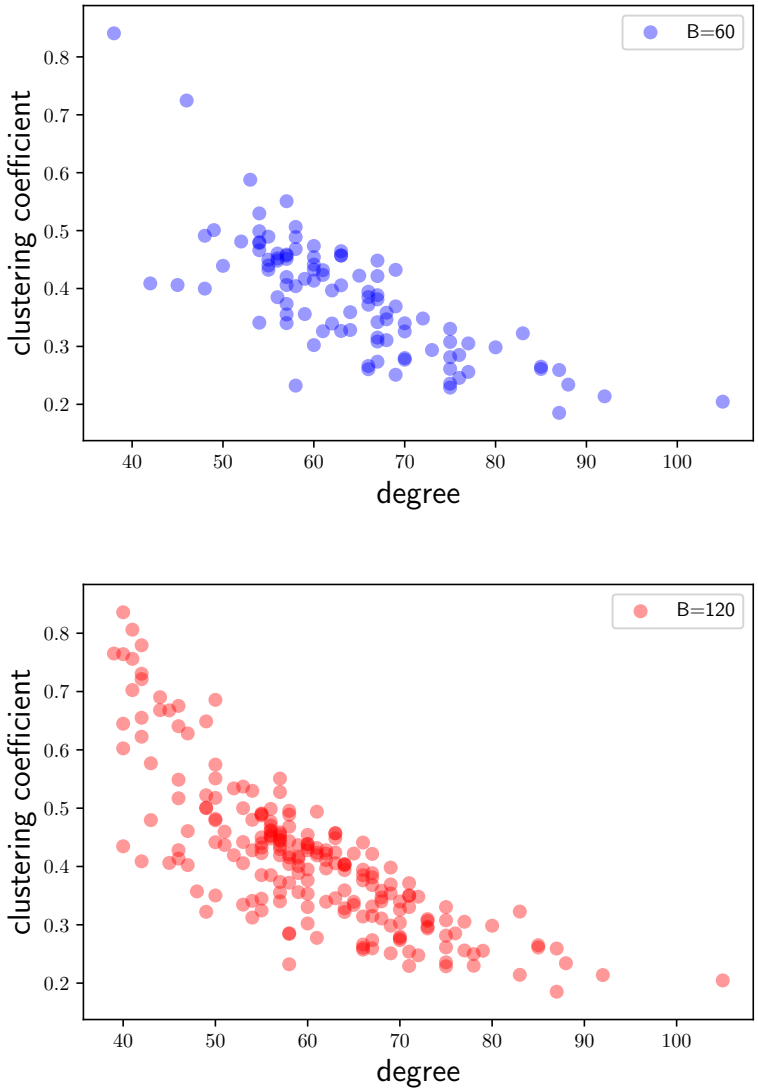


Figure 4.9: Degree vs. Clustering Coefficient of nodes in intervention sets obtained by SAAROUND on Montgomery network. Setting: transmission probability $p = 0.04$, B= 60 (top) vs B = 120 (bottom).

interventions is always performed at time step 0 with budget B_0. As expected, we observe that the number of infections ($\mathbb{E}[\#\text{infections}(\mathbf{X})]$) increases very rapidly with the value of T. This suggests the idea that the earlier the second stage of interventions is performed, the better it will be to contain the epidemic spread. However, we note that this is not always possible, since the vaccines for the second stage may not be available until certain time steps. So, this kind of analysis gives an estimate on the best time frame to get the vaccines ready for the second-stage

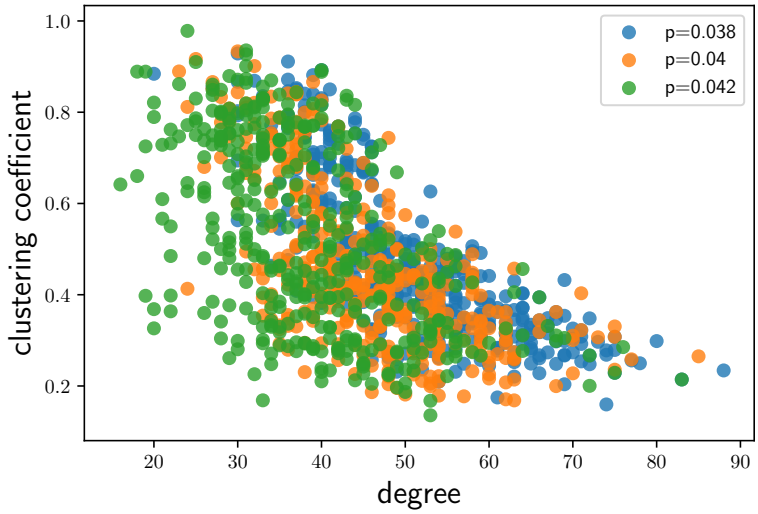


Figure 4.10: Change in node characteristics in intervention set returned by SAAROUND as transmission probability p is varied on Montgomery network.

interventions.

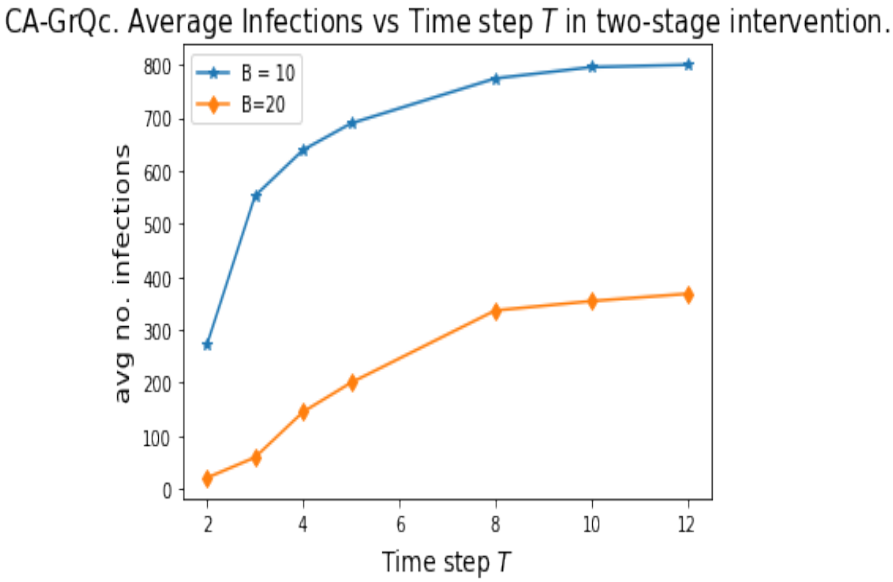


Figure 4.11: Two-stage intervention. Impact of varying the time T of second-stage of intervention on the average number of infections.

Another interesting question is to understand the kind of nodes picked in the two stages of intervention. We expect that the interventions picked in the first stage to be more important, to contain the spread, than those picked in the second stage. But, the question of which network or demographic properties of the nodes

results in the selection at different stages of intervention is still interesting. We examine the degree and age of the nodes in sets picked in each stage.

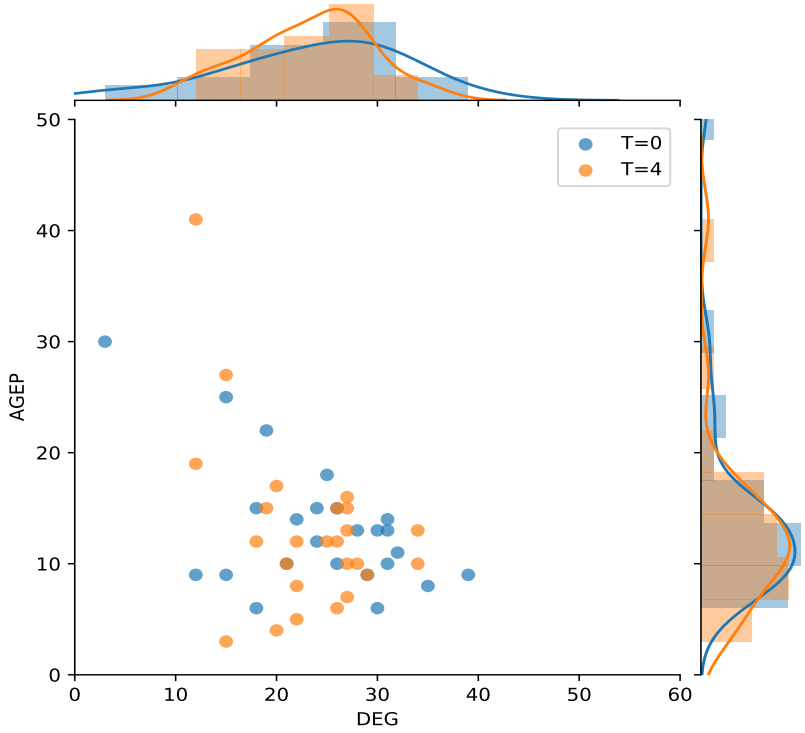


Figure 4.12: Age vs Degree of nodes in the sets $\mathbf{X}_0$ and $\mathbf{X}_4$ in a solution obtained by saaRound for an instance of 2sEpiControl problem. Budget $B = 50$ is divided equally for two-stages, i.e., $B_0 = B_4 = 25$.

Figure 4.12 shows a scatter plot of the node degree and age of the solution to 2sEpiControl with $T = 4$. We observe that there are slight differences between the sets $\mathbf{X}_0$ and $\mathbf{X}_4$: $\mathbf{X}_0$ has slightly higher degree nodes, whereas $\mathbf{X}_4$ has slightly lower age nodes. But more importantly, *it is not the case that all high degree nodes are used in* $\mathbf{X}_0$.

Chapter 5

Robust Interventions for Min-Max Objective

The EpiControl problem assumes that either the source set or the distribution on sources (where the sources can be sampled from a given distribution) is fixed. However, it is to be noted that there might be multiple seeding scenarios of the epidemic model that are feasible. Let $\mathcal{I}$ denote the set of scenarios which capture these multiple possibilities.

In such a setting where multiple scenarios are possible, it is more useful to find *robust* interventions that gives *simultaneous* approximation guarantees over multiple scenarios, instead of an optimal solution for a specific scenario. This can be modeled through a min-max objective, where the goal is to minimize the maximum expected outbreak size in any scenario from set $\mathcal{I}$. We note that the saaRound algorithm is not robust in this sense, and is optimized only for a specific scenario.

The min-max type approach to handle uncertainty has a solid foundation in the field of Stochastic Optimization. For instance, [42] use this approach for robust influence maximization.

5.1 Summary of Results

In this work, we formalize the problem of designing robust intervention strategies, for a given set of scenarios, corresponding to different sources of outbreak. This problem is referred to as problem MinMaxEpiControl. Informally, the goal is to find intervention set $\mathbf{X}$ bounded in size by B, so that the maximum expected outbreak size over a set of scenarios $\mathcal{I}$ is minimized. Also, we consider a two-stage version of the MinMaxEpiControl and show that our algorithm can be adapted as a heuristic for this setting. Our contributions are described below.

1. Approximation algorithms with rigorous guarantees. We designed the mmRound for MinMaxEpiControl by combining a linear programming (LP) rounding approach with the sample average approximation technique from stochastic optimization.

    - **Single-stage, $p = 1$ case**: We designed mmRound algorithm that uses a graph separator subroutine to round the fractional solution from LP to an integral solution. We showed that this algorithm has a significantly better approximation factor of $O(\log |V|)$, instead of logarithmic in the number of paths, as in case of saaRound algorithm from the Chapter 4. [83].
    - **Single-stage, $p < 1$ case.** mmRound uses the randomized rounding in saaRound algorithm and obtains similar guarantees as in that algorithm.

2. Empirical evaluation of algorithms on random graphs. We used our algorithms to compute and analyze interventions for different random graph models. First, we found that there are significant differences in the network properties of nodes picked in the solutions at different times. Further, solutions to the min-max objective are very different from those designed for a specific source distribution.

5.2 Algorithm

We describe our algorithm mmRound, which is based on rounding a linear relaxation of an integer program for the MinMaxEpiControl problem. We first consider the setting where each $S \in \mathcal{I}$ is a deterministic subset of nodes of V when the intervention is done at the beginning and for $p = 1$. Then, we discuss briefly how the extensions to two-stage, the probability distribution over sources, and $p < 1$ case, can be handled; the latter two cases involve using the sample average approximation technique, as in the saaRound [83] algorithm.

5.2.1 Algorithm mmRound for deterministic sources case of MinMaxEpiControl

We start with some definitions needed for the algorithm. Since $p = 1$ in this setting, the SIR outcome for a scenario $S \in \mathcal{I}$ is deterministic. For $S \in \mathcal{I}$, let $\mathcal{P}_{v,S}$ denote the set of paths from any node in S to node $v \notin S$ in G. Let $y_{v,S}$ denote the indicator variable to check whether node v becomes infected (i.e., reachable from S) when the outbreak starts at S. Let x_v denote indicator that node v is vaccinated.

Algorithm mmRound involves the following steps.

1. Solve the following linear program (LP), as described in Lemma 13

$$\min \quad z \quad \text{s.t.} \tag{5.1}$$

$$\forall v, S,\ \forall P \in \mathcal{P}_{v,S} : \sum_{u \in P} x_u \geq 1 - y_{v,S} \tag{5.2}$$

$$\sum_{v} x_v \leq B \tag{5.3}$$

$$\forall S : \sum_{v} y_{v,S} \leq z \tag{5.4}$$

$$\text{All variables} \in [0, 1]. \tag{5.5}$$

2. Let x, y be the optimal fractional solution to (LP). We round it to an integral solution X, Y in the following manner:

   (a) Round $Y_{v,S} = 1$ for each v, S if $y_{v,S} \geq \frac{1}{2}$, otherwise set $Y_{v,S} = 0$.

   (b) Solve the multi-commodity separator problem: find the smallest subset $\mathbf{X}$ of nodes V to remove so that for each $S \in \mathcal{I}$ and each v with $y_{v,S} < \frac{1}{2}$, node v is disconnected from S in $G[V - \mathbf{X}]$, using the algorithm of [34] (also, as described in Lemma 4.2 of [18]).

The integral version of (LP) in mmRound algorithm, i.e., the program with the same constraints, plus all variables being binary, is a valid program. Though (LP) has exponentially many constraints (one for each path), it turns out, (LP) can be solved in polynomial time, as we discuss below.

Lemma 13. *The linear program (LP) in* mmRound *algorithm can be solved in polynomial time.*

Proof. The proof can be showing using the ellipsoid method [36]. Ellipsoid method provides polynomial-time solutions to linear programs even if they have an exponential number of constraints, as is the case with the LP in mmRound algorithm.

The key idea is to construct a separation oracle $\mathcal{O}$, which given a candidate solution x, y, either verifies that it is a feasible solution or finds a constraint that is violated. If the oracle runs in polynomial time, the ellipsoid method too works in polynomial time.

For the linear program (LP) in mmRound, given a candidate solution x, y, it is easy to verify whether or not the objective value is within a certain bound. Further, it is easy to show that the budget constraint is satisfied. However, the main challenge is to verify that for each pair v, S, the constraint for each path P is satisfied, since there can be an exponential number of paths. Our separation oracle instead solves this as the shortest path problem. Let us consider the weight

of a path to be the sum of all the x_u variables for nodes u on that path. Then, the path constraint involves checking whether the path weight is at least $1 - y_{v,S}$. This can be done directly by running the shortest path algorithm using node weights for each scenario $S \in \mathcal{I}$. If it turns out that for some v, S, the distance to v is more than $1 - y_{v,S}$, the shortest path gives us a violated constraint. □

Theorem 14. *Let* $\mathbf{X}$ *denote the vaccination set computed by the* mmRound *algorithm. Let* $\mathbf{X}^*$ *denote an optimal solution to the instance of* MinMaxEpiControl *problem. Then,*

$$\max_{S \in \mathcal{I}} \mathbb{E}[\textbf{\textit{numinf}}(G, S, \mathbf{X})] \leq 2 \max_{S \in \mathcal{I}} \mathbb{E}[\textbf{\textit{numinf}}(G, S, \mathbf{X}^*)],$$

and $|\mathbf{X}| = O(B \log n)$.

Proof. Let x, y, z denote the fractional solution to (LP). The objective value of (LP) is a lower bound on the optimum. This is because, any solution to MinMaxEpiControl, which would be an integral solution, is also a solution to the LP. Therefore, we have $z \leq \max_{\mathbf{s}} \mathbb{E}[\texttt{numinf}(G, S, \mathbf{X}^*)]$. Step 2(b) of the rounding ensures that for each S, if $y_{v,S} = 0$, then node v is disconnected from S in $G[V - \mathbf{X}]$. Therefore,

$$\mathbb{E}[\texttt{numinf}(G, S, V)] \leq \sum_v Y_{v,S} \leq 2 \sum_v y_{v,S} \leq 2z.$$

The bound on $|\mathbf{X}|$ follows from the fact that the fractional solution $2x$ is a separator for all the multi-cut pairs (S, v) with $y_{v,S} \leq 1/2$, since $\sum_{u \in P} 2x_u \geq 2(1 - y_{v,S}) \geq 1$. From [18], it follows that there is an integral separator for all these multi-cut pairs of cost $O(\log n \sum_u x_{u0})$, and therefore, the theorem follows. □

5.2.2 Extension to two-stage version of MinMaxEpiControl

Let x_{u0} be an indicator for node u to be vaccinated at time $t = 0$. Similarly, let x_{uT} be an indicator for u to be vaccinated at time $t = T$. Here, we have to pick a disjoint subset of nodes $\mathbf{X}_0$ and $\mathbf{X}_T$ to vaccinate at times $t = 0$ and $t = T$, respectively. Our algorithm involves the following changes.

1. For each $S \in \mathcal{I}$, run the breadth-first search (BFS) from the nodes in S, and let $V_{S,t}$ be the nodes at level t of the search (with nodes in S being at level 0). Let $V_{S,\geq t} = \cup_{t' \geq t} V_{S,t'}$.

2. We modify constraints (5.2) to the following

$$\sum_{u \in P} x_{u0} + \sum_{u \in P \cap V_{S,\geq T}} x_{uT} \geq 1 - y_{v,S} \tag{5.6}$$

   We also add the constraints $\sum_v x_{vT} \leq B_T$ and $\forall u,\ x_{u0} = 1 - x_{uT}$.

3. Let x, y be the optimal fractional solution to (LP). We round it to an integral solution in the following manner:

   (a) Round $Y_{v,S} = 1$ for each v, S if $y_{v,S} \geq \frac{1}{2}$, otherwise set $Y_{v,S} = 0$.

   (b) For each $v \in V$, and $t = 0, T$, set $X_{vt} = 1$ independently with probability $\min\{1, 2x_{vt} \log(4nN)\}$, where $N = max_{v,S} \mid \cup_{v,S} \mathcal{P}_{v,S}|$.

5.2.3 Extension to probabilistic sources and transmission

Both these cases involve using the sample average approximation technique. We first describe the case of probabilistic sources; the ideas for the $p < 1$ case are similar. Consider a source set $\mathbf{s}_r \in \mathcal{I}$, which specifies a probability distribution over a set of subsets $D = \{\mathbf{s}_r^1, \ldots, \mathbf{s}_r^k\}$, i.e., $\mathbf{s}_r^i$ is the source with probability $p(\mathbf{s}_r^i)$. We introduce variables $y_{v,\mathbf{s}^i,r}$ corresponding to each such set, and add constraints:

$$\sum_{u \in P} x_{u0} + \sum_{u \in P \cap V_{\mathbf{s},\geq T}} x_{uT} \geq 1 - y_{v,\mathbf{s}_r^i},$$

and

$$\sum_i p(\mathbf{s}_r^i) \sum_v y_{v,\mathbf{s}_r^i} \leq z$$

The rest of the steps of the rounding remain the same.

5.3 Experiments

In our experiments, we addressed the following questions.

1. **Properties of the Min-Max Objective.** How does the min-max objective vary depending on the time T of the second stage intervention and the number of interventions available? How much higher is the objective if we use a solution tailored for a specific source distribution $\mathbf{s}$, instead of for the min-max objective?

2. **Properties of nodes selected for intervention at each stage.** What are the characteristics of nodes picked for intervention in both stages of intervention? Are there any network properties in which these nodes differ?

3. **Impact of time T and the ratio B_0/B_T on the effectiveness of two-stage interventions.** How does the benefit of interventions reduce as the second-stage of intervention is delayed? What is the impact of the ratio of budgets allotted in the time steps $t = 0$ and $t = T$ (i.e., B_0/B_T ratio)?

5.3.1 Dataset and Methods

We designed experiments with three very different kinds of randomly generated networks, in order to explore the effect of network structure on the results. We

considered, the small world [53], the preferential attachment [7], and the BTER models [55]. All these networks were used in experiments on saaRound algorithm as well. However, we mainly focused on the BTER graph for most of the experiments in this chapter. Table 5.1 presents the details of these datasets.

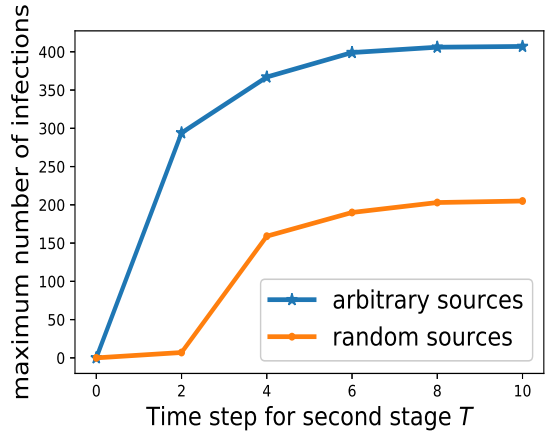
We use mmRound algorithm to solve MinMaxEpiControl on these networks; for a single $\mathbf{s}$ distribution and $p < 1$. This is the same setting as consider in saaRound algorithm [83]. There are no baselines known for MinMaxEpiControl problem , so we use the solution for a specific $\mathbf{s}$ (namely, uniform distribution) for comparison.

Dataset	**Nodes**	**Edges**
Small World (SW)	2500	14833
Preferential (PA)	10000	19996
BTER	5000	35272

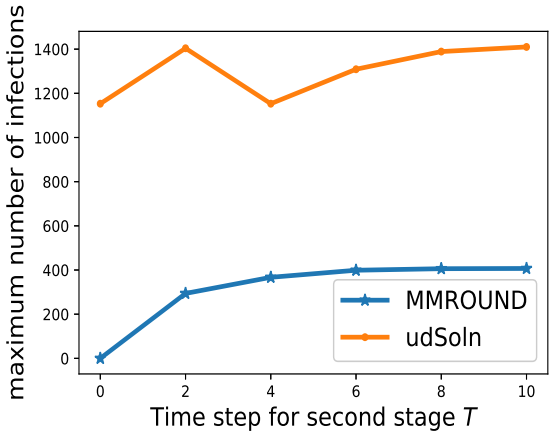
Table 5.1: Description of datasets used in Chapter 5

5.3.2 Properties of Min-Max Objective

We now study the difference between the min-max objective with multiple $\mathbf{s}$ scenarios in set $\mathcal{I}$ vs when there is a single scenario. We would expect the min-max version to become a harder problem and would have a higher objective value. In Figure 5.1 (a), we show how these objectives compare with each other. The blue curve corresponds to the min-max objective value for a given set $\mathcal{I}$ of possible scenarios. The orange curve corresponds to a single $\mathbf{s}$, which is a uniform distribution. We observe that the min-max objective value is much higher, implying that the problem is much harder. In Figure 5.1 (b), we use the solution, referred to as "udSoln", to the $\mathbf{s}$ being a uniform distribution (orange) for the min-max problem, and find that it is much worse than the solution to the min-max objective (blue).



(a)

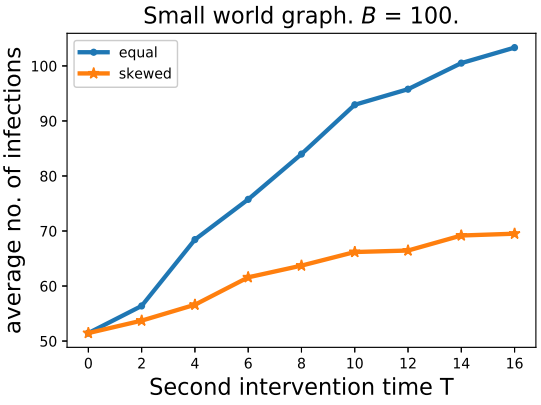


(b)

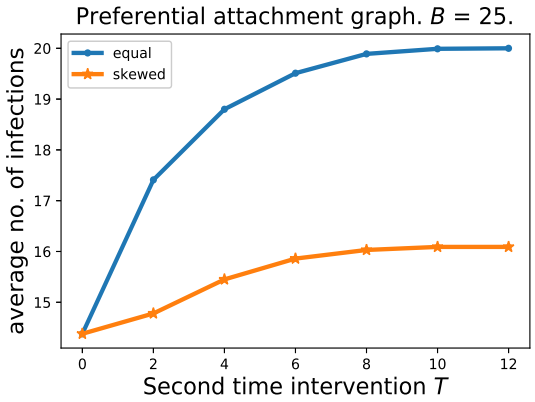
Figure 5.1: (a) Comparison of objective value of MMROUND with arbitrary sources to that with random sources. (b) Comparison of objective value of solution to MINMAXEPICONTROL using MMROUND for arbitrary sources with that of objective value when udnSoln is used as solution to the same problem.

5.3.3 Impact of intervention delay and budgets

We run MMROUND algorithm to compute two-stage interventions on the SW and PA datasets for different choices of the ratio (budget split) B_0/B_T. Figure 5.2 presents the corresponding results. Figure 5.2(a) shows that as we delay the time T for the second stage of intervention, the average number of infections increases, for a fixed budget B. The blue and orange curves correspond to splitting the budget equally and unequally (skewed) among the two stages respectively. As expected, we observe that there is significant benefit in splitting the budget in a



(a)



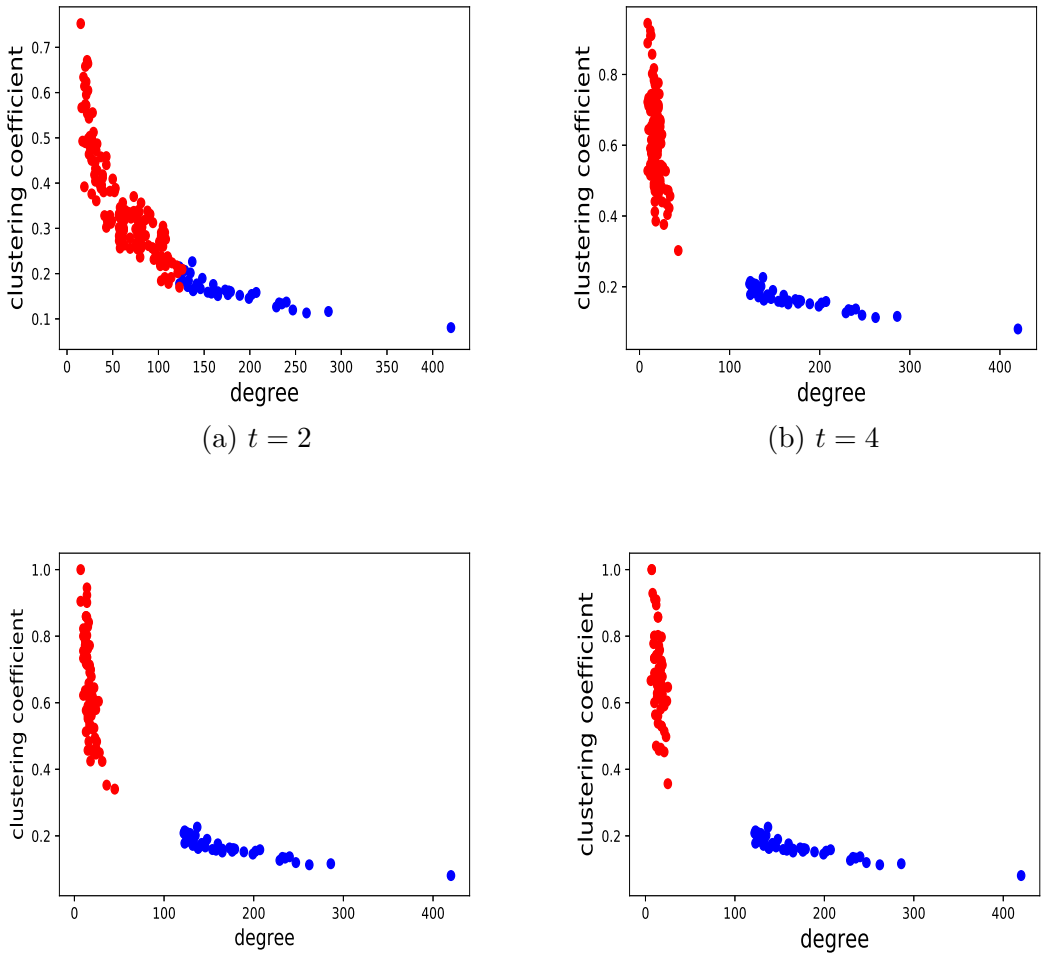
(b)

Figure 5.2: Comparison of two-stage strategies with equal budget ($B_0 = B_T = \frac{B}{2}$), and skewed budgets ($B_0 = \frac{3B}{4}$ and $B_T = B - B_0$). The X-axis corresponds to the time T at which second stage of intervention is performed. The Y-axis corresponds to the average number of infections.

skewed manner, allotting more to time 0 and less to time T, and the number of infections increases with T. Figure 5.2(b) shows that the difference between the two types of budget splits is accentuated when B is smaller.

5.3.4 Characteristics of nodes picked for intervention

In this section, we analyze the characteristics of nodes picked for intervention in each stage of the solution to the two-stage version of MinMaxEpiControl. Specifically, we look at the degree and clustering coefficient of nodes in the solutions. In Figure 5.3, we present the characteristics of nodes picked in the two



(a) $t = 2$ (b) $t = 4$

(c) $t = 6$ (d) $t = 8$

Figure 5.3: BTER graph. Scatter plot. The blue and red points correspond to nodes picked at $T = 0$ and $T = t$ respectively. The X-axis represents the degree, while the Y-axis represents the clustering coefficient of the nodes. $B = 100$.

stages of intervention for different settings. The Figure 5.3 corresponds to budget $B = 100$ and time for second stage of intervention $t = T$ for $T \in \{2, 4, 6, 8\}$. Quite surprisingly, for a fixed budget, as t value increases, the red points corresponding to nodes picked for time t tend to have low degrees and high clustering coefficients, whereas the blue points corresponding to nodes picked at time 0 tend to have high degree and low to moderate clustering coefficients. Although, we observe that the separation between blue and red points is much more pronounced in this setting compared to the two-stage version of EpiControl.

Chapter 6

Group Interventions to Control IAS Spread

The EpiControl problem assumes the SIR model and node-scale interventions. In this work, we considered the MultiPath model, which is a multi-scale epidemiological process on a temporal network modeling the spread of an invasive alien spread across a landscape [61] (described in Chapter 2). In this context, we studied the IAScontrol problem of designing group-scale interventions to minimize the spread under budget constraints and intervention delays. This chapter was a result of collaborative work and the contributions of the collaborators are suitably mentioned.

We designed an integer linear programming based algorithm to find effective group-scale interventions by adapting techniques used in saaRound algorithm approach. We showed rigorous bounds on its performance.

Further, we provided a framework to solve node- or group-scale interventions in the context of other epidemic models that follow SIR-class dynamics such as SI, SEIR, etc.

6.1 Summary of Results

Our contributions are summarized as follows.

- We considered the group-scale intervention problem, IAScontrol [1], to design control strategies for invasive alien species spread in MultiPath model. [61].
- We showed that the IAScontrol problem is NP-hard, even when the graph is a tree. Further, we showed that a variation of this problem (in which the goal is to minimize the cost of the interventions while ensuring that the expected number of infections is bounded) is very hard to approximate due to the group-level interventions. This motivated bicriteria approximations.
- The underlying network of the multi-pathway model is a directed, edge-labeled, and edge-weighted temporal graph. We introduce the concept of time-expanded network [2], which is an explicit representation of the interactions at every time step. We showed that the MultiPath model for a finite time horizon can be provably reduced to a SIR diffusion process on the corresponding time-expanded network.
- We designed SpreadBlocking algorithm for the IAScontrol problem for choosing the groups to intervene, given resource constraints and delay in intervening. Our method uses a combination of the sample average approximation (SAA) technique along with linear relaxation and rounding. We showed rigorous guarantees on its performance.
- We studied the performance of our algorithm on five real-world networks, which are considered in [61]. We showed that the performance of our algorithm is consistently superior compared to the popular baselines for variations in model parameters, seeding scenarios, budget, and intervention delay. [3]

[1] This problem is considered along with our collaborators [90]
[2] This is a collaborative effort [90]
[3] The experimental study is a collaborative effort. Particularly, the implementation of the baselines and the simulator are the contributions of our collaborations. The implementation of SpreadBlocking is a contribution of this dissertation.

6.2 Hardness of IAScontrol and Bicriteria approximations

We first show that the IASCONTROL problem is **NP**-hard even when the network G is a tree. The approximation hardness for IASCONTROL is still open. We show that the IASCONTROLMINBUDGET variation is very hard to approximate due to the group level decisions.

Lemma 15. *The* IASCONTROL *problem is NP-complete even when G is a tree.*

Proof. Our reduction is from a variation of the Unbalanced Graph Cut problem [41]: Given a graph $G = (V, E)$, a source node s, and cost c_v for each node $v \in V$. The goal is to choose a subset $V' \subset V$ of nodes such that $\sum_{v \in V'} c_v \leq B$, and $|\{u : u \text{ is reachable from } s \text{ in } G - V'\}|$ is minimized. By modifying the reduction in [41] (the authors in this work consider the edge version of the problem), it can be shown that the above variation is also NP-hard even for the case when G is a tree.

We observe that the MULTIPATH model generalizes the SI epidemic process on a graph, by considering a single pathway in which each node is in a singleton group. We reduce the above variation of the Unbalanced Graph Cut problem to the IASCONTROL on a tree, with the model parameters chosen accordingly so that the MULTIPATH corresponds to the SI process. Taking $T = |V|$, the number of infections is equal to the number of nodes reachable from S in the residual graph after the intervened set of nodes are removed. □

Lemma 16. *It is NP-hard to approximate the* IASCONTROLMINBUDGET *to within an $O(2^{\log^{1-o(1)} n/2})$ factor.*

Proof. Our reduction is from the node version of the Label Cut problem [102], which is defined in the following manner: given a graph $G = (V, E)$, a source node s, a sink node t, a label $\ell(v) \in L$ for each node $v \in V$, and a cost c_i for

each label $i \in L$, the objective is to choose a subset $L' \subset L$ such that the nodes s and t are disconnected in $G - \{v : \ell(v) \in L'\}$, and $\sum_{i \in L'} c_i$ is minimized. The hardness result of [102] (who consider the edge labeled version of the problem) can be modified to show that the same hardness holds for the node version of Label Cut as well.

We reduce the node version of the Label Cut problem to an instance G' of IAScontrolMinBudget. G' is basically the same as G, with an additional set U of n nodes connected to the node t (so that the total number of nodes is $2n$). We consider the set L to be the groups $\mathcal{Q}$, and choose parameters so that there is a single pathway in MultiPath, which corresponds to the SI model. We choose $K = n$. Then, the $\inf_{\mathrm{T}}(G, \mathbf{s}, \tau_{\mathrm{d}}, \{v \mid g(v) \in \mathcal{Q}^*\}) \leq K$ if and only if the node t is disconnected from s when the nodes in $\{v : g(v) \in \mathcal{Q}^*\}$ are removed, else the number of infections will be at least N, since all the nodes in U will be infected if t is infected. Thus, a solution $\mathcal{Q}^*$ to the IAScontrolMinBudget instance on G' corresponds to a min-label s,t cut in G. Since the number of nodes in G' is $2n$, the hardness result follows from the bound of [102]. □

Bicriteria approximation. The hardness in Lemma 16 motivates the notion of bicriteria approximation: we say that a solution $\mathcal{Q}'$ is an (α, β)-approximation if

$$\inf_{\mathrm{T}}(G, \mathbf{s}, \tau_{\mathrm{d}}, \{v \mid g(v) \in \mathcal{Q}'\}) \leq \alpha \inf_{\mathrm{T}}(G, \mathbf{s}, \tau_{\mathrm{d}}, \{v \mid g(v) \in \mathcal{Q}^*\})$$

and $\sum_{Q_q \in \mathcal{Q}'} c_q \leq \beta B$, where $\mathcal{Q}^*$ is an optimal solution with $\sum_{Q_q \in \mathcal{Q}^*} c_q \leq B$.

6.3 Approach for IAScontrol

First, we show that the MultiPath can be represented as a SIR process on an auxiliary network called the time-expanded network. We note that this idea can be extended to other related SIR models such as SI, SEIR, etc. Next, we present

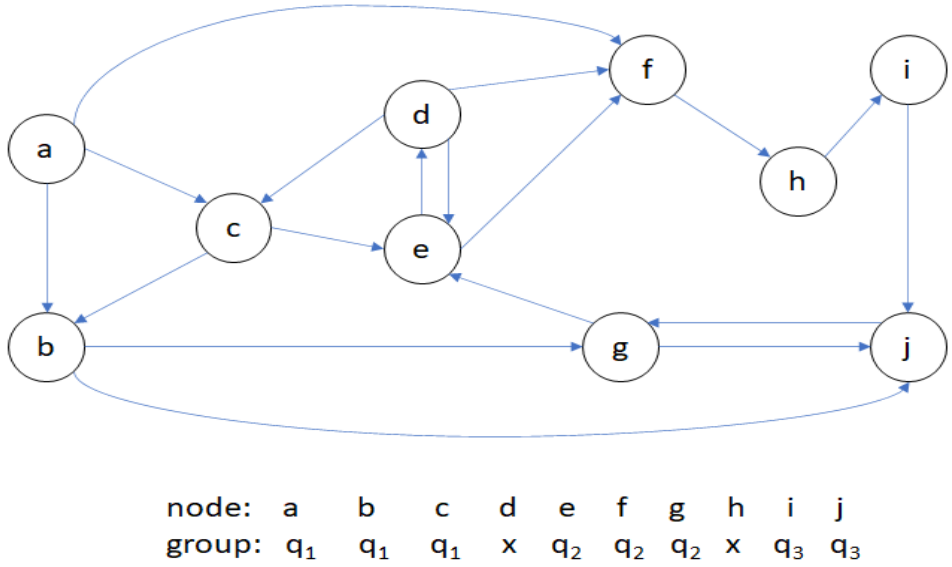


Figure 6.1: An example network showing nodes and the associated groups. Nodes d and h are not associated with any group denoted by x, therefore, they are not eligible for group-scale interventions.

a group intervention algorithm for the IASCONTROL problem using SAA and LP rounding techniques, and show its guarantees.

6.3.1 Time-expanded network

In this section, we represent the MULTIPATH model as a SIR process (instead of the SEI process) on another network called the time-expanded network.

Let $H_{\text{te}}(V_{\text{te}}, E_{\text{te}})$ denote the time-expanded network corresponding to the multi-pathway model on $G(V, E)$. The key idea is to treat every node u at each time step as a distinct node, i.e., we have $T+1$ copies $\{u_0, \ldots, u_T\}$ of node u, where u_i represents the copy of u at time step i. To incorporate the exposed state in the underlying SEI process of the multi-pathway model, we have ℓ additional copies $\{u_{i,0}, \ldots, u_{i,\ell-1}\}$, corresponding to each u_i, where ℓ is the latency period. The edge set E_{te} consists of exactly the following four types of edges which corresponds to different events in a SEI process:

- $(v_i, u_{i+1,0}, \lambda, i)$, $\forall (v, u, \lambda, i) \in E$ with weight $w(v, u, \lambda, i)$

(captures $\mathbf{S} \to \mathbf{E}$ through pathway λ)

- $(u_{i,r}, u_{i,r+1})$ for $r \in [0, \ell - 2]$ (captures $\mathbf{E} \to \mathbf{E}$)
- $(u_{i,\ell-1}, u_{i+\ell})$ (captures $\mathbf{E} \to \mathbf{I}$)
- (u_i, u_{i+1}) (captures $\mathbf{I} \to \mathbf{I}$)

All edges of types other than $\mathbf{S} \to \mathbf{E}$ have weight 1. For the special case of the SI diffusion process ($\ell = 0$), there are no nodes of the form $u_{i,r}$ and it has two edge types, $\mathbf{I} \to \mathbf{I}$ as defined above and $\mathbf{S} \to \mathbf{I}$: $(v_i, u_{i+1,0}, \lambda)$ with weight $w(v, u, , \lambda, i)$. An example time-expanded network is shown in Figure 6.2 for a single pathway which corresponds to a simulation instance on G in Figure 6.1.

Let $\sigma_G(v, t)$ be the state of a vertex v in G at time t, which can be either $\mathbf{S}$, $\mathbf{E}$, or $\mathbf{I}$. Similarly, let $\sigma_{H_{\text{te}}}(u_i, t)$ (resp. $\sigma_{H_{\text{te}}}(u_{i,r}, t)$) be the state of a vertex u_i (resp. $u_{i,r}$) in H_{te} at time t with $\mathbf{S}$, $\mathbf{I}$, and $\mathbf{R}$ being the possible states. Let $\mathcal{O}_G$ denote a stochastic disease outcome of the SEI model on G – this specifies the state $\sigma_G(v, t)$ for each (v, t), and set of the edges (u, v, λ, t) such that node u infects v at time t through pathway λ. Similarly, let $\mathcal{O}_{H_{\text{te}}}$ denote a disease outcome in the SIR model on H_{te}. We say that $\mathcal{O}_{H_{\text{te}}}$ is consistent with $\mathcal{O}_G$ if:

(i) for any u, $\sigma_G(u, i) = \mathbf{I}$ (resp. $\sigma_G(u, i) = \mathbf{S}$) $\iff$ $\sigma_{H_{\text{te}}}(u_i, i) = \mathbf{I}$ (resp. $\sigma_{H_{\text{te}}}(u_i, i) = \mathbf{S}$);

(ii) $\sigma_G(u, i + r) = \mathbf{E}$, $r \in [0, \ell - 1]$) $\iff$ $\sigma_{H_{\text{te}}}(u_{i,r}, i + r) = \mathbf{I}$;

(iii) $\sigma_G(u, i - 1) = \sigma_G(u, i) = \mathbf{I}$ $\iff$ $\sigma_{H_{\text{te}}}(u_{i-1}, i) = \mathbf{R}$; and (iv) u infects v on edge (u, v, λ, i) in G at time i $\iff$ node u_{i-1} infects node $v_{i,0}$ at time i on edge (u_{i-1}, v_i, λ).

Given $\mathcal{O}_G$, for a time t, let $\mathcal{O}_G(\leq t)$ be a snapshot of $\mathcal{O}_G$ up to time step t. Similarly, $\mathcal{O}_{H_{\text{te}}}(\leq t)$ is a snapshot of $\mathcal{O}_{H_{\text{te}}}$ up to t time steps.

Example for time-expanded network construction.: We consider a simplified version of MultiPath for the sake of explaining the construction of a time-expanded network. Figure 6.1 shows a directed network with nodes and

their associated groups. Figure 6.2 shows the time-expanded that corresponds to a simulation instance of MULTIPATH on this network.

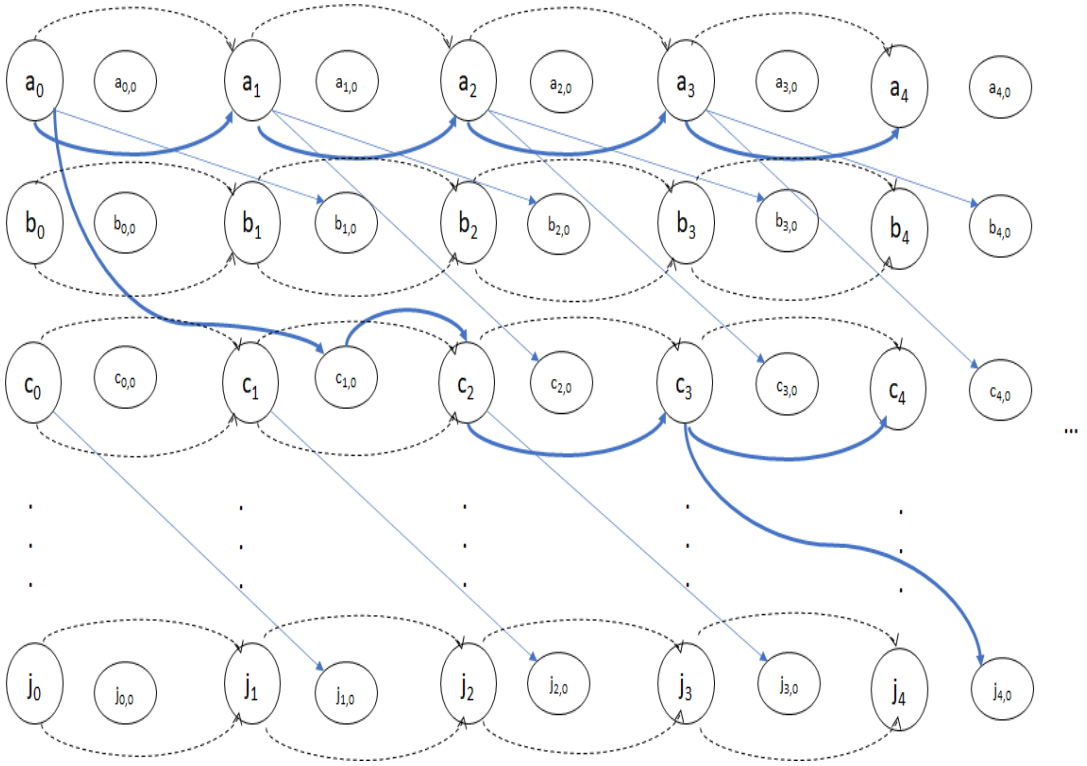


Figure 6.2: A (partial) time-expanded graph corresponding to the following simulation instance of MULTIPATH on G in Figure 6.1 for latency period $\ell = 1$: Node a is the seed infection. At time step $t = 1$, node a infects c and node c infects node j at time $t = 3$. The edges in bold correspond to *live edges* which are the subset of the edges (bold, dashed, and dotted) that correspond to the events in SEI process.

Theorem 17. *Consider the multi-pathway diffusion process on $G(V, E)$ for T time steps with a latency period $\ell \geq 0$ and the SIR process on the corresponding time-expanded graph $H_{te}(V_{te}, E_{te})$. Then, for any outcome $\mathcal{O}_G$ and a consistent outcome $\mathcal{O}_{H_{te}}$, the probability that $\mathcal{O}_G$ is the outcome in the multi-pathway process on G is equal to the probability that $\mathcal{O}_{H_{te}}$ is the outcome in the SIR process on H_{te}.*

Proof. The proof is by induction on time t. We will assume that the latency period $\ell > 0$. The proof for $\ell = 0$ (corresponding to SI process) can be shown using the same approach. At $t = 0$, by definition, $\forall v \in V(G)$ we have $\sigma_G(v, 0) = \mathbf{I} \iff$

$\sigma_{H_{\text{te}}}(v_0, 0)) = \mathbf{I}$ and $\sigma_G(v, 0) \neq \mathbf{E}$. Suppose that $\Pr(\mathcal{O}_G(\leq t)) = \Pr(\mathcal{O}_{H_{\text{te}}}(\leq t))$ for some $t \geq 0$.

For the induction step, we consider all events that can occur at time $t+1$ w.r.t. a node v in $\mathcal{O}_G$ at time $t+1$ on a case by case basis. It is enough to prove that for each such event, the corresponding event in $\mathcal{O}_{H_{\text{te}}}$ for $t+1$ has the same probability. This is because, for a given time instance, any event corresponding to v is independent of events corresponding to any other node v' at time $t+1$, since the state of v at time $t+1$ depends only on the system state at time t.

Case 1. $\mathbf{I} \rightarrow \mathbf{I}$ Consider the event where v is in state $\mathbf{I}$ in $\mathcal{O}_G$ at t and it remains in this state at time $t+1$. By model definition, the probability of this event is 1. In $\mathcal{O}_{H_{\text{te}}}$, the corresponding event is,

(i) $\sigma_{H_{\text{te}}}(v_t, t) = \mathbf{I}$,

(ii) $\sigma_{H_{\text{te}}}(v_{t+1}, t) = \mathbf{S}$, and

(iii) $\sigma_{H_{\text{te}}}(v_{t+1}, t+1) = \mathbf{I}$. By induction assumption, (i) and (ii) are true. Since the weight on edge (v_t, v_{t+1}) is 1, this event happens with probability 1.

Case 2. $\mathbf{S} \rightarrow \mathbf{S}$ Suppose v is in state $\mathbf{S}$ in $\mathcal{O}_G$ at t and it remains in this state at time $t+1$. For this to occur, v should not be infected through any edge of the form $(v', v, \lambda, t+1)$ at time $t+1$. Let $E(v, \mathcal{O}_G, t+1)$ denote the set of such edges. By model definition, the probability of this event is $\prod_{(v',v,\lambda,t+1)\in E(v,\mathcal{O}_G,t+1)} \big(1 - w(v', v, \lambda, t+1)\big)$. In $\mathcal{O}_{H_{\text{te}}}$, the corresponding event is,

(i) $\sigma_{H_{\text{te}}}(v_t, t) = \mathbf{S}$ and

(ii) $\sigma_{H_{\text{te}}}(v_{t+1}, t+1) = \mathbf{S}$. By induction assumption, (i) is true.

By definition of H_{te}, probability that $v_{t+1,0}$ is infected by the edge $(v'_t, v_{t+1,0}, \lambda, t+1)$ is $w(v', v, \lambda, t+1)$. This is equal to $\prod_{(v',v,\lambda,t+1)\in E(v,\mathcal{O}_G,t+1)} \big(1 - w(v', v, \lambda, t+1)\big)$.

Case 3. $\mathbf{S} \rightarrow \mathbf{E}$ Suppose v is in state $\mathbf{S}$ at time t and is infected at time $t+1$ through one or more edges from $E(v, \mathcal{O}_G, t+1)$. Let E' denote the set of such

edges. The probability of this event occurring is $\prod_{(v',v,\lambda,t+1)\in E'} w(v',v,\lambda,t+1)$. The corresponding event in $\mathcal{O}_{H_{\text{te}}}$ is,

(i) $\sigma_{H_{\text{te}}}(v_t,t) = \mathbf{I}$,

(ii) $\sigma_{H_{\text{te}}}(v_{t+1,0},t) = \mathbf{S}$, and

(iii) $v_{t+1,0}$ is infected at time $t+1$ through edges in the set $E'' = \{(v'_t, v_{t+1,0}, \lambda, t+1) \mid (v',v,\lambda,t+1) \in E'\}$.

By induction assumption, (i) and (ii) are true. By definition of H_{te}, probability that $v_{t+1,0}$ is infected by the edge $(v'_t, v_{t+1,0}, \lambda, t+1)$ is $w(v',v,\lambda,t+1)$. Therefore, the probability that $v_{t+1,0}$ is infected through all the edges in E' is $\prod_{(v',v,\lambda,t+1)\in E'} w(v',v,\lambda,t+1)$.

Case 4. $\mathbf{E} \to \mathbf{E}$ Consider the event where v is in state $\mathbf{E}$ in $\mathcal{O}_G$ at t and it remains in this state at time $t+1$. By model definition, this can happen only if v was infected at $t-r$ for some $r \in [0, \ell-1]$. Under this assumption, the probability of this event is 1. In $\mathcal{O}_{H_{\text{te}}}$, the corresponding event is,

(i) $\sigma_{H_{\text{te}}}(v_{t-r,r},t) = \mathbf{I}$,

(ii) $\sigma_{H_{\text{te}}}(v_{t-r,r},t) = \mathbf{S}$, and

(iii) $\sigma_{H_{\text{te}}}(v_{t-r,r+1},t+1) = \mathbf{I}$.

By induction assumption, (i) and (ii) are true. Since the weight on edge $(v_{t-r,r}, v_{t-r,r+1})$ is 1, this event happens with probability 1 as well.

Case 5. $\mathbf{E} \to \mathbf{I}$ Consider the event where v is in state $\mathbf{E}$ in $\mathcal{O}_G$ at t and transitions to state $\mathbf{I}$ at time $t+1$. By model definition, this can happen only if v was infected at $t-\ell$. Under this assumption, the probability of this event is 1. In $\mathcal{O}_{H_{\text{te}}}$, the corresponding event is,

(i) $\sigma_{H_{\text{te}}}(v_{t-\ell,\ell},t) = \mathbf{I}$,

(ii) $\sigma_{H_{\text{te}}}(v_{t+1},t) = \mathbf{S}$, and

(iii) $\sigma_{H_{\text{te}}}(v_{t+1},t+1) = \mathbf{I}$. By induction assumption, (i) and (ii) are true. Since the weight on edge $(v_{t-\ell,\ell}, v_{t+1})$ is 1, this event happens with probability 1.

For the special case of the SI diffusion process ($\ell = 0$), the proof follows by replacing Case 3 ($\mathbf{S} \to \mathbf{E}$) with ($\mathbf{S} \to \mathbf{I}$) and ignoring Cases 4 and 5. □

6.3.2 Group Intervention Algorithm

SpreadBlocking (Algorithm 2) is based on the sample average approximation (SAA) technique from stochastic optimization. Let $\{H^1, \ldots, H^M\}$ be the set of M simulation outcomes corresponding to SIR process on H_{te}, where each $H^j = (V_{\text{te}}, E^j_{\text{te}})$, such that $E^j_{\text{te}} \subseteq E_{\text{te}}$. We solve a linear relaxation of the IAS-control problem, restricted to these samples, and the resulting objective value is guaranteed to be close to the actual expected number of infections. Table 6.1 summarizes the quantities and variables used in the linear program, referred to as $LP_{\tau_{\text{d}}}$.

Term	**Definition**
M	Number of simulation outcomes
$S_{te} \subseteq V_{\text{te}}$	Fixed set of sources of infection $\forall H^j$
$\mathcal{R}(H^j) \subseteq V_{\text{te}}$	Set of nodes in H^j reachable from S_{te} via a directed path
$x_{q,\tau_{\text{d}}} = 1$	if group $Q_q \in \mathcal{Q}$ is intervened at time-step τ_{d}
$y^j_{u,i} = 1$	if $u_i \in V_{\text{te}}$ is infected in H^j at time-step i (there is a directed from S_{te} to u_i in H^j), i.e., $\sigma_{H_{\text{te}}}(u_i, i) = \mathbf{I}$.
$y^j_{u,i,r} = 1$	if $u_{i,r}$ is infected in H^j at time-step i (there is a directed from S_{te} to $u_{i,r}$ in H^j), i.e, $\sigma_{H_{\text{te}}}(u_{i,r}, i) = \mathbf{I}$
$z^j_u = 1$	if node u_i or $u_{i,r}$ is infected in H^j (corresponds to u being infected within T in G)
g_m	maximum number of groups to which the set of nodes on any path in any H^j belong to. Typically, we expect $g_m << k$

Table 6.1: Notation for SpreadBlocking algorithm

Let $\mathcal{Q}' \subseteq \mathcal{Q}$ be any intervention set for τ_{d}. Let $V_{\text{te}}(\mathcal{Q}') = \{v_i, v_{i,r} \in V_{\text{te}} \mid g(v) \in \mathcal{Q}' \text{ and } i \geq \tau_{\text{d}}\}$, be the set of nodes in H^j to which intervention $\mathcal{Q}'$ applies. Let $V(\mathcal{Q}') = \{v \in V \mid v_i, v_{i,r} \in V_{\text{te}}(\mathcal{Q}')\}$ be the set of nodes in G to which intervention $\mathcal{Q}'$ applies. Let $H^j - V_{\text{te}}(\mathcal{Q}')$ denote the subgraph of H^j induced by removing all nodes in $V_{\text{te}}(\mathcal{Q}')$ from H^j. Let $I^j(\mathcal{Q}') = \{v \in V | \ \exists i \text{ s.t. } v_i \text{ or } v_{i,r} \in$

Algorithm 2 SpreadBlocking algorithm

Input: $G = (V, E)$, set of sources $S \subseteq V$, budget B, time horizon T, intervention delay τ_{d}

Output: intervention set $\mathcal{Q}_{\mathrm{SB}} \subseteq \mathcal{Q}$

1: Construct time expanded network H_{te} from G

2: Construct M simulations of the SIR process $\{H^1 = (V_{\mathrm{te}}, E^1_{\mathrm{te}}), \ldots, H^M = (V_{\mathrm{te}}, E^M_{\mathrm{te}})\}$ with $S_{te} = \{u_0 \mid u \in S\}$ as sources on the time-expanded network H corresponding to SEI process on G (as described in Section 6.3.2)

3: Solve the linear program $LP_{\tau_{\mathrm{d}}}$ defined as follows:

$$
\begin{aligned}
(LP_{\tau_{\mathrm{d}}})\ \min \quad & \frac{1}{M}\sum_{j}\sum_{u} z^j_u \\
\forall i < \tau_{\mathrm{d}}, u_i, u_{i,r} \in \mathcal{R}(H^j) \quad : \quad & y^j_{u,i} = 1, y^j_{u,i,r} = 1 \\
\forall u_i, u_{i,r} \in \mathcal{R}(H^j) \quad : \quad & z^j_u \geq y^j_{u,i},\ z^j_u \geq y^j_{u,i,r} \\
\forall (v_{i-1}, u_{i,0}) \in E^j_{\mathrm{te}} : y^j_{u,i,0} \quad \geq \quad & y^j_{v,i-1} - x_{g(u),\tau_{\mathrm{d}}} \\
\forall (u_{i,r}, u_{i,r+1}) \in E^j_{\mathrm{te}} : y^j_{u,i,r+1} \quad \geq \quad & y^j_{u,i,r} - x_{g(u),\tau_{\mathrm{d}}} \\
\forall (u_{i-\ell,\ell-1}, u_i) \in E^j_{\mathrm{te}} : y^j_{u,i} \quad \geq \quad & y^j_{u,i-\ell,\ell-1} - x_{g(u),\tau_{\mathrm{d}}} \\
\forall (u_{i-1}, u_i) \in E^j_{\mathrm{te}} : y^j_{u,i} \quad \geq \quad & y^j_{u,i-1} - x_{g(u),\tau_{\mathrm{d}}} \\
\forall i \geq \tau_{\mathrm{d}}, \forall u_i, u_{i,r} \in \mathcal{R}(H^j) \quad : \quad & y^j_{u,i} \leq 1 - x_{g(u),\tau_{\mathrm{d}}} \\
& y^j_{u,i,r} \leq 1 - x_{g(u),\tau_{\mathrm{d}}} \\
\sum_{Q_q \in \mathcal{Q}} x_{q,\tau_{\mathrm{d}}} \quad \leq \quad & B \\
\text{All variables} \quad \in \quad & [0,1]
\end{aligned}
$$

4: (Rounding) Let $\mathbf{x}, \mathbf{y}, \mathbf{z}$ be the optimal fraction solution to $LP_{\tau_{\mathrm{d}}}$. Round it to an integral solution X, Y, Z using the following rounding procedure: (i) For each H^j, u_i, set $Y^j_{u,i} = 1$ if $y^j_{u,i} \geq \frac{1}{2}$. Similarly, for each $H^j, u_{i,r}$, set $Y^j_{u,i,r} = 1$ if $y^j_{u,i,r} \geq \frac{1}{2}$. (ii) For each $H^j, u \in V$, set $Z^j_u = 1$ if $z^j_u \geq \frac{1}{2}$. (iii) For each $Q_q \in \mathcal{Q}$, set $X_{q,\tau_{\mathrm{d}}} = 1$ if $x_{q,\tau_{\mathrm{d}}} \geq \frac{1}{2g_m}$ where g_m, such that $g_m \leq k$ is the maximum number of groups associated with the set of nodes on any path in any H^j.

5: **return** $\mathcal{Q}_{\mathrm{SB}} = \{Q_q \mid X_{q,\tau_{\mathrm{d}}} = 1\}$

$\mathcal{R}(H^j - V_{\mathrm{te}}(\mathcal{Q}'))\}$ denote the number of infections (nodes still reachable from S_{te} in H^j) in V. Let $I(\mathcal{Q}') = \frac{1}{M}\sum_j I^j(\mathcal{Q}')$ denote the average number of infections in V restricted to the M simulations.

Let $\hat{\mathcal{Q}}_{opt} = \operatorname{argmin}_{\mathcal{Q}''} I(\mathcal{Q}'')$ be an intervention set that achieves the minimum average number of infections on the simulations. Then, let $I_{opt} = \inf_{\mathrm{T}}(V(\mathcal{Q}^*))$,

i.e, the expected number of infections achieved by an optimal solution $\mathcal{Q}^*$ to the given instance of the IASCONTROL.

We first show that the average number of infections $I(\mathcal{Q}')$ achieved by any intervention set $\mathcal{Q}'$ restricted to the M simulations is close to the expected number of infections $\inf_{\mathrm{T}}(\mathcal{Q}')$ for that intervention set.

Lemma 18. *Let the number of groups $|\mathcal{Q}| = k \geq 2$. If $M \geq 24nk \log k$, with probability at least $1 - \frac{1}{k}$, for any intervention set $\mathcal{Q}' \subseteq \mathcal{Q}$, we have $I(\mathcal{Q}') \in [\frac{1}{2}\inf_{\mathrm{T}}(V(\mathcal{Q}')), \frac{3}{2}\inf_{\mathrm{T}}(V(\mathcal{Q}'))]$.*

Proof. From equivalence in Section 6.3.1, we have,

$$\mathbb{E}[I(\mathcal{Q}')] = \mathbb{E}[I^j(\mathcal{Q}')] = \inf_{\mathrm{T}}(V(\mathcal{Q}')) \tag{6.1}$$

The $I^j(\mathcal{Q}')$ variables are independent and $\frac{I^j(\mathcal{Q}')}{n} \in [0,1]$ where $|V| = n$ is the number of nodes in G. Using Chernoff bound in Theorem 1.1 of [23] to $M\frac{I^j(\mathcal{Q}')}{n}$, we have

$$\Pr\left(\frac{I^j(\mathcal{Q}')}{n} \notin \left[\frac{M}{2n}\inf_{\mathrm{T}}(V(\mathcal{Q}')), \frac{3M}{2n}\inf_{\mathrm{T}}(V(\mathcal{Q}'))\right]\right) \leq 2exp(-\frac{M}{12n}\inf_{\mathrm{T}}(V(\mathcal{Q}'))) \tag{6.2}$$

Since, there is at least one infection (sources are assumed to be infected as they cannot be intervened), we have $\inf_{\mathrm{T}}(V(\mathcal{Q}')) \geq 1$. This probability is at most $2e^{-2k\log k} = \frac{2}{k^{2k}}$. The number of possible intervention sets $\mathcal{Q}'$ is at most 2^k (there is a one-to-one mapping between a group $Q \in \mathcal{Q}$ and associated $V(\{Q\})$). Therefore, for $M = 24nk\log k$, the probability that there exists an intervention set $\mathcal{Q}' \subseteq \mathcal{Q}$ such that $I(\mathcal{Q}') \notin [\frac{1}{2}\inf_{\mathrm{T}}(V(\mathcal{Q}')), \frac{3}{2}\inf_{\mathrm{T}}(V(\mathcal{Q}'))]$ is at most $2^k\frac{2}{k^{2k}} \leq \frac{1}{k}$ for $k \geq 2$. □

Let $ILP_{\tau_{\mathrm{d}}}$ denote the integral version of $LP_{\tau_{\mathrm{d}}}$, i.e., with all variables required to be in $\{0,1\}$. Below, we show that $ILP_{\tau_{\mathrm{d}}}$ is *valid*

Lemma 19. *For every feasible intervention set $\mathcal{Q}' \subseteq \mathcal{Q}$, there exists a feasible integral solution $\bar{\mathbf{x}}, \bar{\mathbf{y}}, \bar{\mathbf{z}}$ to ILP_{τ_d} such that $\frac{1}{M}\sum_j \sum_u \bar{z}_u^j = I(\{Q_q : x_{q,\tau_d} = 1\})$. If $\bar{\mathbf{x}}, \bar{\mathbf{y}}, \bar{\mathbf{z}}$ is an optimal solution to ILP_{τ_d}, $I(X_{opt}) = \frac{1}{M}\sum_j \sum_j \bar{z}_u^j$.*

Proof. Given a feasible intervention set $\mathcal{Q}'$. Let us define $x_{q,\tau_\mathrm{d}} = 1$ for each Q_q if $Q_q \in \mathcal{Q}'$ and $\mathbf{x}_{q,\tau_\mathrm{d}} = 0$ otherwise. Let us define $y_{u,i}^j = 1$ (resp. $y_{u,i,r}^j = 1$) if $u_i \in \mathcal{R}(H^j - V(\mathcal{Q}'))$ (resp. $u_{i,r} \in \mathcal{R}(H^j - V(\mathcal{Q}'))$ (set of nodes reachable from S_{te} can be computed by a BFS from S_{te}). Now, define $z_u^j = 1$ if $\exists$ some i s.t. $y_{u,i}^j = 1$ or $y_{u,i,r}^j = 1$. We have,

$$I^j(\mathcal{Q}') = |\mathcal{R}(H^j - V(\mathcal{Q}'))| = \sum_j z_u^j \tag{6.3}$$

Consider a path P from some $s_0 \in S_{te}$ to a node $u_{i,0} \in \mathcal{R}(H^j - V(\mathcal{Q}))$. For the edge $(v_{i-1}, u_{i,0}) \in E_{\mathrm{te}}^j$, where $i \geq \tau_\mathrm{d}$, by construction, we have $y_{u,i,0}^j \geq y_{v,i-1}^j - x_{g(u),\tau_\mathrm{d}}$ (since $y_{u,i,0}^j$ =1) for every v_{i-1} that has a directed edge into node $u_{i,0}$, which implies that the corresponding edge-constraint in LP_{τ_d} is satisfied. Now, consider a node $u_{i,0} \notin \mathcal{R}(H^j - V(\mathcal{Q}))$ and $i \geq \tau_\mathrm{d}$, if $u_{i,0}$ has an edge from a node in $\mathcal{R}(H^j - V(\mathcal{Q}))$, it must be that $g(u) \in \mathcal{Q}'$, otherwise $u_{i,0}$ would be infected (i.e, reachable from S_{te} by this path). This implies that $x_{g(u),\tau_\mathrm{d}} = 1$ and the constraint is satisfied.

With similar arguments, we can show that the other constraints are satisfied. Since $\mathcal{Q}'$ is a feasible intervention set to the IAScontrol problem, we have $|\mathcal{Q}'| \leq B$. The budget constraint in ILP_{τ_d} is satisfied as $\sum_{Q_q \in \mathcal{Q}} x_{q,\tau_\mathrm{d}} \leq B$ by construction.

The converse follows with a similar argument. Additionally, we will need the property that if $y_{u,i,0}^j = 1$ for node $u_{i,0}$, then there is a path P from S_{te} to some node $u_{i,0}$ with all y variables corresponding to nodes on path having value 1. This is satisfied by the edge constraints in LP_{τ_d} (adding all the constraints on the edges on path P gives this constraint). □

Lemma 20. *For any H^j, and any node $u_i \in V_{te}$ with $y^j_{u,i} < \frac{1}{2}$ (resp. for $u_{i,r} \in V_{te}$ with $y^j_{u,i,r} < \frac{1}{2}$), rounding in* SpreadBlocking *algorithm ensures that the node u_i (resp. $u_{i,r}$) is not reachable from S_{te} in $H^j - V_{te}(\mathcal{Q}_{\text{SB}})$, where $\mathcal{Q}_{\text{SB}}$ is the intervention set computed by the algorithm.*

Proof. Let $\mathcal{P}_{u_i,j}$ be the set of paths from S_{te} to node u_i in H^j. Let $U_{\tau_\text{d}}(P)$ denote the set of nodes on path P at distance τ_d or more from S_{te}. Let us denote $G_{\tau_\text{d}}(P) \subseteq \mathcal{Q}_{\text{SB}}$ to be the groups to which nodes in $U_{\tau_\text{d}}(P)$ belong.

We will prove the statement for $u_i \in V_{\text{te}}$ (a similar argument works for $u_{i,r}$ case). Given $y^j_{u,i} < \frac{1}{2}$, by rounding in SpreadBlocking we have $Y^j_{u,i} = 0$ (i.e., u_i is not infected in H^j). A node is uninfected in H^j if and only if for every path $P \in \mathcal{P}_{u_i,j}$ at least one group in $G_{\tau_\text{d}}(P)$ is in $\mathcal{Q}_{\text{SB}}$, i.e., u_i is not reachable from S_{te} in $H^j - V(\mathcal{Q}_{\text{SB}})$. This corresponds to the constraint $\sum_{Q_q \in G_{\tau_\text{d}}(P)} x_{q,\tau_\text{d}} \geq 1 - y^j_{u,i} > \frac{1}{2}$ (this path-based constraint could be obtained from the edge constraints in LP_{τ_d} by adding all the constraints on the edges on path P).

Assume for the sake of contradiction that none of the groups in $Q_q \in G_{\tau_\text{d}}(P)$ have $x_{q,\tau_\text{d}} \geq \frac{1}{2g_m}$. Then, we have $\sum_{Q_q \in G_{\tau_\text{d}}(P)} x_{q,\tau_\text{d}} < \frac{1}{2}$ as there could be at most g_m groups for any path where each $Q_q \in G_{\tau_\text{d}}(P)$ has $x_{q,\tau_\text{d}} < \frac{1}{2g_m}$. This is a contradiction as a feasible solution to LP_{τ_d} satisfies this constraint. Therefore, for a node $u_i \in H^j$, on each path $P \in \mathcal{P}_{u_i,j}$ there exists some group $Q_q \in G_{\tau_\text{d}}(P)$ with $x_{q,\tau_\text{d}} \geq \frac{1}{2g_m}$ implying that some $Q_q \in G_{\tau_\text{d}}(P)$ is in $\mathcal{Q}_{\text{SB}}$. □

Lemma 21. *Let $\mathcal{Q}_{\text{SB}} = \{Q_q \mid X_{q,\tau_d} = 1\}$ be the intervention set computed by* SpreadBlocking *algorithm, then we have $|\mathcal{Q}_{\text{SB}}| \leq 2g_m B$.*

Proof. The rounding procedure in SpreadBlocking scales each x_{q,τ_d} variable by a factor at most $2g_m$. Therefore,

$$|\mathcal{Q}_{\text{SB}}| = \sum_{Q_q \in \mathcal{Q}} X_{q,\tau_\text{d}} \leq \sum_{Q_q \in \mathcal{Q}} 2g_m x_{q,\tau_\text{d}} \leq 2g_m B \tag{6.4}$$

The first inequality follows from the rounding and the second inequality follows from the budget constraint in $LP_{\tau_{\mathrm{d}}}$. □

Theorem 22. *Let $M \geq 24nk \log k$. Let $\mathcal{Q}_{\mathrm{SB}}$ be the intervention set computed by* SpreadBlocking *algorithm. Then with probability $1 - \frac{1}{k}$, $\inf_{\mathrm{T}}(V(\mathcal{Q}_{\mathrm{SB}})) \leq 6 \inf_{\mathrm{T}}(V(\mathcal{Q}^*))$ where $\mathcal{Q}^* \subseteq \mathcal{Q}$ is an optimal solution for the given instance of* IAScontrol*, and $|\mathcal{Q}_{\mathrm{SB}}| \leq 2g_m B$.*

Proof. By Lemma 20, our rounding ensures that any node $u_i \in V_{\mathrm{te}}$ (resp. $u_{i,r} \in V_{\mathrm{te}}$) with $y^j_{u,i} < \frac{1}{2}$ (resp. $y^j_{u,i,r} < \frac{1}{2}$) will be disconnected from S_{te} in H^j. Since, for any $u \in V$, $z^j_u \geq y^j_{u,i}, y^j_{u,i,r}$ for all $i \leq T$, $z^j_u \geq \frac{1}{2}$ if there exists some i such that $y^j_{u,i} \geq \frac{1}{2}$ or some (i,r) $y^j_{u,i,r} \geq \frac{1}{2}$. Therefore, from Lemma 20 it follows that, node u is disconnected from S_{te} in H^j if $Z^j_u = 0$.

Then, for all H^j,

$$I^j(\mathcal{Q}_{\mathrm{SB}}) = |\{u \mid Z^j_u = 1\}| = \sum_u Z^j_u \leq \sum_{u: z^j_u \geq \frac{1}{2}} 2z^j_u \leq 2 \sum_u z^j_u \tag{6.5}$$

Then,

$$I(\mathcal{Q}_{\mathrm{SB}}) = \frac{1}{M} \sum_j I^j(\mathcal{Q}_{\mathrm{SB}}) \leq \frac{1}{M} \sum_{u,j} 2z^j_u \leq 2I(\hat{\mathcal{Q}}_{opt})$$

The last inequality follows since the $LP_{\tau_{\mathrm{d}}}$ solution is a lower bound on $I(\hat{\mathcal{Q}}_{opt})$. Furthermore, we have $I(\mathcal{Q}_{\mathrm{SB}}) \leq 2I(\hat{\mathcal{Q}}_{opt}) \leq 2I(\mathcal{Q}^*)$, by definition of $\hat{\mathcal{Q}}_{opt}$. By Lemma 18, with probability $1 - \frac{1}{k}$ we have $I(\mathcal{Q}^*) \leq \frac{3}{2} \inf_{\mathrm{T}}(V(\mathcal{Q}^*))$ and $\frac{1}{2} \inf_{\mathrm{T}}(V(\mathcal{Q}_{\mathrm{SB}})) \leq I(\mathcal{Q}_{\mathrm{SB}})$. This implies $4I(\mathcal{Q}^*) \leq 6 \inf_{\mathrm{T}}(V(\mathcal{Q}^*))$ and $\inf_{\mathrm{T}}(V(\mathcal{Q}_{\mathrm{SB}})) \leq 2I(\mathcal{Q}_{\mathrm{SB}})$. By Lemma 21, we have $|\mathcal{Q}_{\mathrm{SB}}| \leq 2g_m B$. Putting all this together with probability $1 - \frac{1}{k}$, we have

$$\inf_{\mathrm{T}}(V(\mathcal{Q}_{\mathrm{SB}})) \leq 2I(\mathcal{Q}_{\mathrm{SB}}) \leq 4I(\mathcal{Q}^*) \leq 6 \inf_{\mathrm{T}}(V(\mathcal{Q}^*)) \tag{6.6}$$

and $|\mathcal{Q}_{\mathrm{SB}}| \leq 2g_m B$. □

6.4 Framework to extend saaRound approach to other epidemic models

Based on the ideas developed in the approach used for IAScontrol, particularly, the results of the equivalence theorem (Theorem 17), we formulate a framework to extend our approach to other epidemic models that follow SIR (or SEIRS) class dynamics. We present this simple framework in this section.

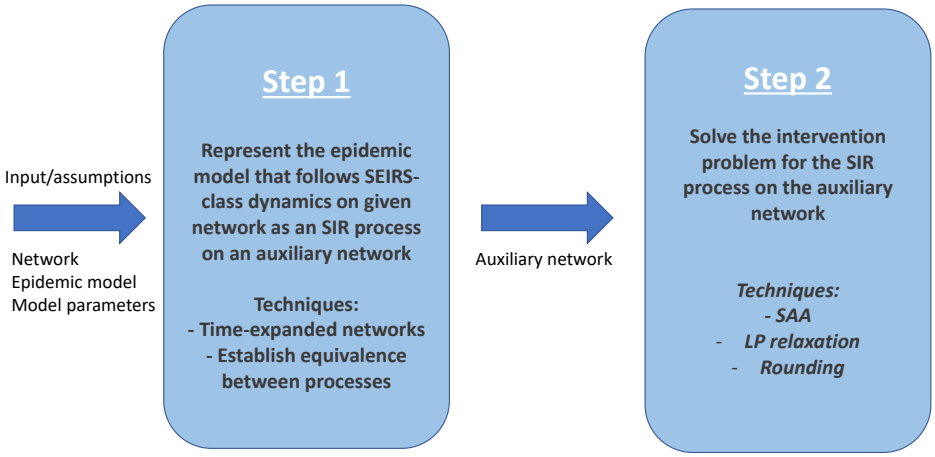


Figure 6.3: Framework for controlling spread in epidemic models that follow SIR class dynamics.

Our framework to solve control problems on complex epidemic models that follow SIR-class dynamics consists of the following two main steps:

(Step 1.) Represent the dynamics of the given epidemic model as a SIR process. This can be achieved by the notion of auxiliary graphs such as time-expanded networks. Show the equivalence between the process (corresponding to the given epidemic model) on the given network and a SIR process on the auxiliary graph constructed in this step.

(Step 2.) Solve the problem of designing interventions for the SIR process on the auxiliary graph generated in Step 1. This can be solved by adapting the saaRound approach. Compute M sampled outcomes of the SIR process on this

auxiliary graph. Formulate an Integer Linear Program (ILP) for the intervention problem on the M samples. Solve the corresponding LP relaxation and round the fractional optimal solution thus obtained to obtain the intervention set.

6.5 Experiments

We conducted experiments on several real-world networks. The main objective of our experiments (relevant to this dissertation) was to evaluate the performance of SpreadBlocking.

We compared our algorithm against popular baselines with respect to effectiveness in minimizing the spread. We studied its performance relative to the LP solution (which gives a lower bound on optimum for the instance) to compare with the approximation ratios established in Section 7.2.

Also, we evaluated our algorithm by comparing it to the targeted (or node) intervention case, where each node belongs to its own distinct group.

Datasets. Table 6.2 presents a summary of all networks used in our experiments. These were constructed by McNitt et al. [61] and are publicly accessible. We used the values 2 and 500 for the distance function exponent and cut-off respectively. These were among the best model parameters obtained after calibration in their work.

Each network has groups containing, on average, about 20–30 nodes capturing key urban and producing areas. For most datasets (each corresponding to data from a country), a significant portion of the nodes does not belong to any group. Therefore, such nodes will not be considered for the interventions. However, these nodes together cover less than 20% of the total production and population in each country.

Table 6.2: List of networks used and their attributes.

net.	name	nodes	edges	groups	gp. edges
BD	Bangladesh	211	6846	7	141
ID	Indonesia	3296	110640	35	2181
PH	Philippines	673	20108	16	450
TH	Thailand	738	27666	5	48
VN	Vietnam	503	16746	15	426

Experimental setup. The parameter values of the multi-pathway model were chosen to cover the best models with the highest fit to ground truth. The parameters used in our experiments are stated in the plots, for a detailed list of suitable parameters (a full factorial design of the experimental study), we refer to [90].

Each simulation was run for $T = 24$ time steps corresponding to a time-horizon of two years. Seeding scenarios were picked from McNitt et al. [61]. We also added a few more seeding scenarios for more counterfactual experiments. For performance evaluation, we used the mean number of infections across simulation instances as the metric for time horizon $T = 24$.

We compared SpreadBlocking to several baselines for different B and τ_{d} values. In each case, the average fraction of nodes infected was used as the metric for evaluation. The following baselines are used for comparison with our algorithm:

1. *Maximum outflow*: This method corresponds to ordering groups by *outflow* in the group-to-group network (similar to the degree-based method for undirected graphs). Then, pick the top B groups with highest maximum outflow. This method is often used in the invasive species literature [61, 70]. In this case, we considered the annual outflow by aggregating the outflows across different months.

2. *Vulnerability* [83]: This method is based on idea in experimental study of saaRound algorithm. We observed that the number of nodes infected in each group at time step 12 (i.e., halfway in our time horizon) and picked B most vulnerable groups.

Both these baselines were implemented by our collaborators.

We also considered the objective value of the relaxed ILP corresponding to

SpreadBlocking. Recall that since SpreadBlocking is a bi-criteria approximation algorithm, for a given budget B, the solution of the algorithm uses a budget $B' \geq B$. We considered two objective values of the relaxed program:

(i) LP-LB (LB for lower bound) is the objective value corresponding to budget B' and

(ii) LP-BCA (BCA for bi-criteria approximation) is the objective value corresponding to budget B.

LP-LB serves as a lower bound for the expected number of infections for the integral solution, which uses a budget of B'. LP-BCA is used to evaluate the algorithm with respect to the bounds established in Theorem 22.

Rounding schemes. In SpreadBlocking algorithm, the x_{q,τ_d} variables in the fractional optimal solution to LP_{τ_d} are scaled by a factor $2g_m$, and rounded to 1 if the resulting value after the scaling is at least 1. The term g_m corresponds to the maximum number of groups associated with nodes on any path in any sampled graph H^j. To obtain the value of g_m is computationally intensive. Therefore, we use some small constant c as a "surrogate" for the factor $2g_m$. Then, for each value of c we will have a different rounding scheme. Considering $c \in \{3, 4, 8\}$, we notice that the budget violation is higher for higher values of c. Therefore, as c increases, the number of infections tends to be low as well. We found the best balance in this trade-off between budget violation and minimization of the number of infections at $c = 4$. We used this value throughout our experiments.

Performance evaluation. Representative results for two networks are in Figure 6.4, one corresponding to increasing budget B for a fixed τ_d and the other corresponding to increasing intervention delay τ_d for a fixed B. The number of infections for unmitigated spread is also plotted. We observed consistently superior performance of SpreadBlocking algorithm against the baselines. In particular, for lower τ_d, SpreadBlocking performed much better than other strategies.

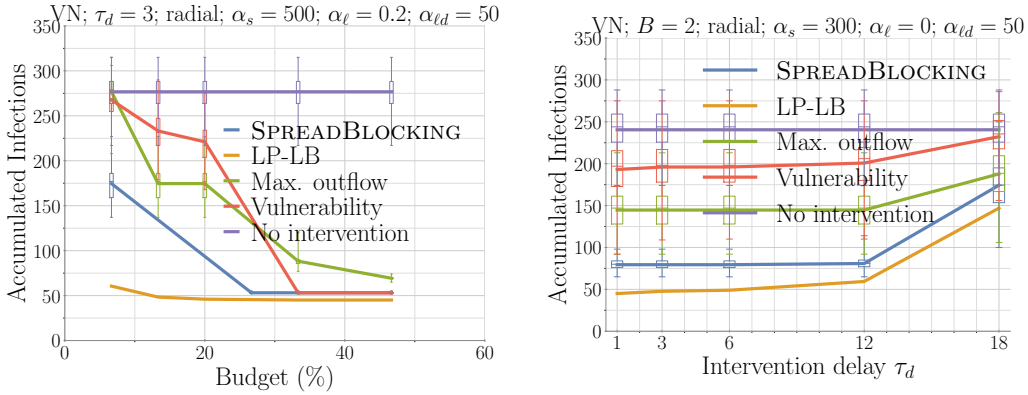


Figure 6.4: Comparison of algorithm with respect to budget and intervention delay. Some representative plots are given. The titles contain the following information in the order in which they are mentioned: network, budget/delay, seeding scenario, and pathway parameters.

We note that for lower budgets and delay, the performance was better suggesting that the solutions provided by SpreadBlocking for early intervention were significantly better than given by other schemes. However, at lower budget, we also noticed that the difference between LP-LB and our algorithm is high, motivating further study of rounding techniques for low budget instances. To further analyze across various networks, model parameters, and seeding scenarios, we compared the ratio of mean infections corresponding to SpreadBlocking and that for each baseline. These results are presented in Figure 6.5(a). We observed that in most cases SpreadBlocking performed significantly better than Max. overflow. This is because the maximum overflow strategy does not account for seeding scenarios. We also observed that it is better than the vulnerability-based strategy indicating that it is not always advisable to intervene only at localities that are at high risk of invasion.

Comparison with LP solution. In Figure 6.5(a), we compared the intervention benefit obtained with that of LP-LB and LP-BCA. Compared with LP-BCA, we observed that SpreadBlocking had much better approximation guarantees in practice, which is also much better than the bounds in Theorem 22. For most cases, the approximation factor (w.r.t LP-BCA) was less than 1.5. Even when

compared to the lower bound LP-LB, we noticed that for almost all cases, the approximation factor was around 1.6 indicating that the performance is near-optimal for the considered networks and scenarios. Figure 6.5(b) corresponds to budget violation given by the ratio of the budget of the solution provided by the algorithm to the given budget. Since SpreadBlocking is a bi-criteria approximation algorithm, the solution provided can violate the budget constraints. We observed this phenomenon in all these experiments as well. In most cases, the budget violation was at most 2. This ratio went down further with the increase in budget. Occasionally, we also observed that the given budget is higher than what is required, leading to a solution with fewer groups as intervention set.

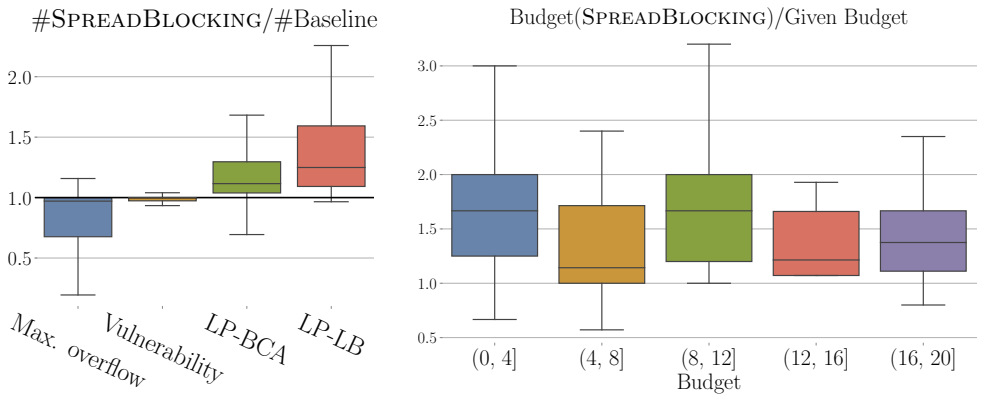


Figure 6.5: (a) Summary of performance of SpreadBlocking across networks, model parameters, seeding scenarios, budget and intervention delay. (b) Budget violation with respect to user given budget B.

Comparison with targeted intervention. The goal in this experiment was to assess group-scale interventions with better performing yet difficult-to-implement individual-based (or targetted) interventions. In Figure 6.6, we compared the two types of interventions for one country. Since each group on average has around 20 nodes, for the sake of comparison, we expressed the results for the group-scale intervention in terms of the number of nodes intervened at (# groups $\times$ avg. nodes per group in the network). We observed that the performance of group-scale interventions was comparable to individual-based interventions.

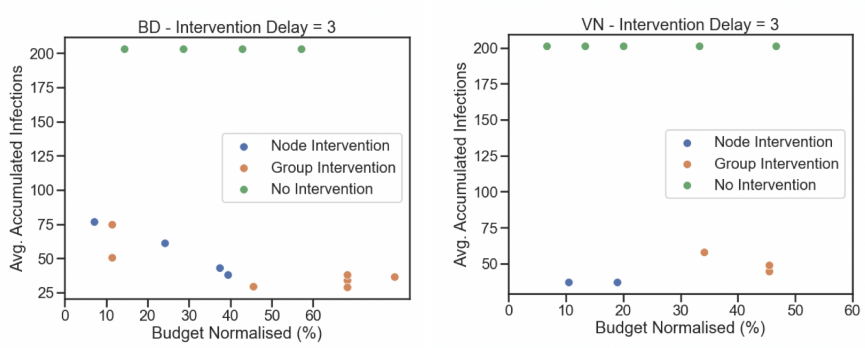


Figure 6.6: Comparison of group-based and individual-based interventions for the parameter set: $\alpha_s \in 300$, $\alpha_\ell \in 0.2$, $\alpha_{\ell d} \in 50$, Moore range $r_{\mathrm{M}} = 1$, start month $= 5$.

Computation time and scalability. SpreadBlocking algorithm scaled well for all the networks considered in our experiments. However, for certain instances of BD, it took longer ($\approx$ 15 mins), due to the solution space as well as the number of infections resulting in the simulations. The main bottleneck in this algorithm too is solving $LP_{\tau_{\mathrm{d}}}$, but using pruning techniques reduced $LP_{\tau_{\mathrm{d}}}$ program size, thereby speeding up the algorithm.

Chapter 7

Scalable Algorithms for EpiControl Problem

This chapter presents scalable algorithms for the EpiControl problem. We identified that using the LP solver is the main bottleneck in saaRound algorithm as the linear program will have $n + nM$ variables and $\sum_j |E_j|$ constraints. We overcome this bottleneck by using the Multiplicative Weights Update (MWU) method along with the sample average approximation (SAA) technique to approximately solve the LP. Further, we provided a memory-efficient version of this algorithm that allows it to scale to large networks — corresponding to country-size populations — with over 300 million nodes and 30 billion edges.

7.1 Summary of Results

- We designed MwuRound algorithm that substantially improves the approach of [83] for finding near-optimal vaccination strategies in networked SIR models. (Section 7.2). This algorithm relies on a subroutine, LSearch-Saa which adapts the multiplicative weights update (MWU) and the sample average approximation techniques to compute an approximate solution to the LP relaxation of the problem instance. By a careful implementation

of MWU, and exploiting the structure of samples, we were able to further improve the running time by an order of magnitude.

- We designed a scalable and memory-efficient version of MwuRound, referred to as MwuRound-Scalable. This algorithm doesn't store all the samples for SAA in memory and instead runs the MWU computations on random samples computed on the fly. We showed that this is an improvement over MwuRound-Scalable both in terms of memory and runtime.

- We evaluated the performance of our methods on a number of real and synthetic networks. We showed that our algorithm scales to a national scale network containing over 334M nodes; runs in a couple of days without memory issues. Further, we showed that these methods have good approximation guarantees in practice, thus providing a highly scalable approach for designing interventions in networked SIR models.

7.2 Algorithm

Notation	**Definition**
$H_j = (V_{H_j}, E_{H_j})$	augmented sampled graph
$a(u, j)$	copy of node u, referred to as a stub, attached to u in H_j
$A(j)$	set of stub nodes in H_j
$\mathcal{P}_{v,j}$	set of paths from S to stub $v = a(u, j)$ in H_j
$\mathcal{P}_j$	set of paths from S to all stubs in $A(j)$
$\mathcal{P}$	$= \bigcup_{j \in [M]} \mathcal{P}_j$, i.e., set of all paths
$\ell(u)$	length of node u
$\ell(v)$ for $v = a(u, j)$	length of a stub node $a(u, j)$
$\ell(P)$ for $P \in \mathcal{P}$	sum of lengths of nodes (including stub node) in path P
$z(P)$	flow on path P

Table 7.1: Summary of notation for MwuRound algorithm

We improve the saaRound algorithm [83] presented in Chapter 4 by bypassing the need to use a solver to directly solve the LP. Instead, we adapt the technique of the Multiplicative weight [4] to find a near-optimal solution to the LP. It will

be easier to present the LP in a slightly different form. Let y_{vj} be an indicator whether the node v gets infected in the sampled outcome H'_j. Let $\mathcal{P}'_v$ be the set of paths from S to node v in any outcome H'_j for $j \in [M]$.

$$(LP_{path}) \qquad \mathrm{Z}_{LP} = \min \frac{1}{M} \sum_j \sum_{v \in V_{H'_j}} y_{vj} \; s.t. \tag{7.1}$$

$$\forall v \in V_{H'_j} \setminus S, \forall P \in \mathcal{P}'_v \, , \sum_{u \in P} x_u + y_{vj} \geq 1 \tag{7.2}$$

$$\sum_{u \in V} x_u \leq B \tag{7.3}$$

$$x_u, y_{vj} \in [0, 1] \tag{7.4}$$

By adding up the constraints for each edge LP_{saa} on a given path, it can be verified that we get the constraint (7.2) of LP_{path}, which is summarized below. This is summarized in the following observation.

Observation 23. The above LP is equivalent to LP_{saa}.

Main ideas and steps. Algorithm MwuRound (Algorithm 5) is our main algorithm; it uses Algorithm LSearch-Saa (Algorithm 4) which in turn uses Algorithm MwuSaa (Algorithm 3) as a subroutine. The main ideas underlying the algorithm are summarized below.

1. Lagrangian multiplier for budget: The dual of LP_{path} is complicated due to a negative coefficient associated with the budget constraint (7.3). We simplify it by changing the objective to $\frac{1}{M} \sum_j \sum_{v \in V_{H'_j}} y_{vj} + \lambda \sum_{u \in V} x_u$, with the multiplier λ for the cost of the solution. The budget constraint is dropped; we refer to this LP as LP_{LM}. This simplifies the resulting LP, since it only has covering constraints. As λ increases, $\sum_u x_u$ will decrease in the optimal solution. Since values for λ are not known a priori, a binary search can be employed to find a suitable value such that $\sum_u x_u \leq B$, which is done in Algorithm LSearch-Saa. The x', y' values returned by Algorithm 3 for λ' provides an approximate solution to LP_{saa}.

2. Constructing augmented sampled graphs. For simplifying the presentation, we construct M sampled graphs $H_j = (V_{H_j}, E_{H_j})$ in the following manner: H_j is initially the same as H'_j, constructed as in the first step of saaRound. Let $A(j) = \{a(u,j) : u \in V_{H'_j} - S\}$, where $a(u,j)$ denotes a copy of node u in H_j, and is referred to as a *stub*. Let $\mathbf{A} = \bigcup_{j\in[M]} A(j)$ be set of all stubs. Each *stub* $a(u,j)$ is attached to u by an edge $(u, a(u,j))$. Overloading the definitions, let $\mathcal{P}_{v,j}$ denote the set of paths from S to a *stub* node $v = a(u,j) \in A(j)$ in H_j. Let $\mathcal{P}_j = \bigcup_{v\in A(j)} \mathcal{P}_{v,j}$ and $\mathcal{P} = \bigcup_{j\in[M]} \mathcal{P}_j$.

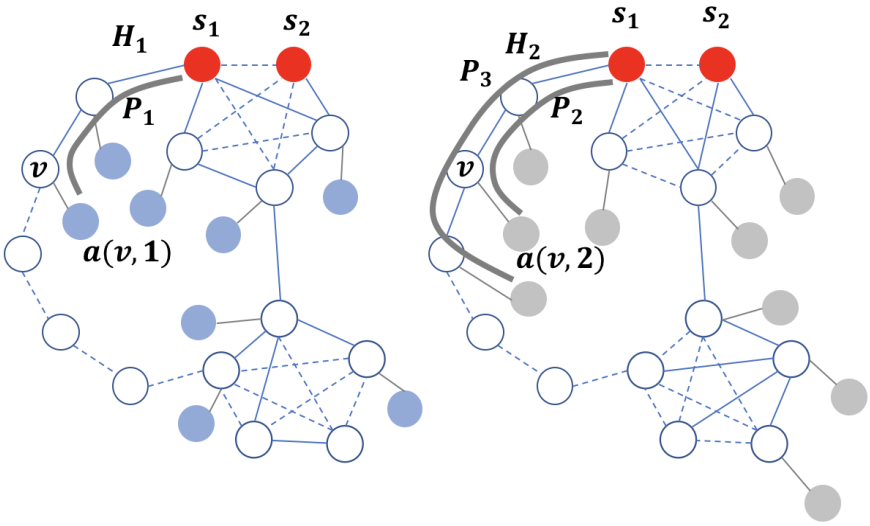


Figure 7.1: Example showing two samples H_1, H_2, and stub nodes denoted by $a(v,j)$ where v is a node in G and j refers to the ID of sample H_j.

3. Variables and costs. We associate a length to each node $u \in (V - S) \bigcup \mathbf{A}$ denoted by $\ell(u)$. For a node $u \in V$, $\ell(u)$ will correspond to the variable x_u, while for a node $v = a(u,j) \in \mathbf{A}$, $\ell(v)$ will correspond to the variable y_{uj}. The length of any path in $P \in \mathcal{P}$ is given by the sum of lengths of nodes on this path. Let $\ell = \langle \ell(u) : u \in V \bigcup \mathbf{A} \rangle$ denote the vector of length variables. Let $c(u) = \lambda$ for $u \in V \setminus S$ denote its capacity, whereas $c(u) = 0$ for $u \in S$. Let $c(v) = \frac{1}{M}$ for $v = a(u,j) \in \mathbf{A}$ denote the capacity of a *stub* v. We will keep track of flows on the network; let $z(P)$ denote the flow on the path $P \in \mathcal{P}$. Let $\mathbf{z} = \langle z(P) : P \in \mathcal{P} \rangle$ denote the vector of flow variables.

4. Simplified LP. Based on the above discussion, we will be solving the following LP, denoted by $LP_\ell(\lambda)$

$$\begin{aligned} \mathrm{Z}_{LR}(\lambda) = \min \sum_u \ell(u)c(u) \qquad & s.t. \\ \forall P \in \mathcal{P} \sum_{u \in P - S} \ell(u) &\geq 1 \\ \forall u : \qquad \ell(u) &\geq 0 \end{aligned}$$

5. Incremental computation of $\ell(\cdot)$***.*** Algorithm MwuSaa computes an approximate solution to LP_ℓ, using the multiplicative weight update technique [4]. It starts by initializing the length $\ell(v) = \delta$ for each $u \in (V \setminus S) \bigcup \mathbf{A}$, where δ has a very small value determined in the analysis. The $\ell(v)$ for $v \in S$ is initialized to zero. Also, for each $u \in V - S$, we set a capacity $c(v) = \lambda$, whereas for $v \in \mathbf{A}$ the capacity $c(v) = \frac{1}{M}$. In each iteration r of the algorithm, and for each augmented sampled graph H_j, we update the lengths of nodes on the paths in $\mathcal{P}_j$ corresponding to an augmented sampled graph H_j until all the paths in $\mathcal{P}_j$ have length at least $\delta(1+\epsilon)^r$ — this value is referred to as $threshold(r)$ for the r^{th} iteration. The algorithm terminates after $r_{max} = \lfloor \log_{1+\epsilon} \frac{1+\epsilon}{\delta} \rfloor$ iterations. Since, the $threshold(r_{max})$ for the r_{max} iteration is in $[\log_{1+\epsilon} \frac{1+\epsilon}{\delta} - 1, \log_{1+\epsilon} \frac{1+\epsilon}{\delta}]$, we are guaranteed that, at termination, all paths in $\mathcal{P}$ are of length in range $[1, 1+\epsilon]$, thereby satisfying the constraints of the linear program.

7.2.1 Analysis

First, we show below that Algorithm MwuSaa gives a $(1+4\epsilon)$-approximate solution to $LP_\ell(\lambda)$ (Theorem 27). The number of iterations and the total running

Algorithm 3 MwuSaa (λ)

Input: parameter λ (we assume the network $G = (V, E), S$, subgraphs $H_1, \ldots, H_M$, ϵ are fixed,
$\delta = (1+\epsilon)((1+\epsilon)L)^{-\frac{1}{\epsilon}}$ where L is the maximum number of nodes on any path in G)

Output: ℓ

Initialize $\ell(u) = \delta$ for all $u \in (V - S) \cup \mathbf{A}$,
$z(P) = 0$ for all $P \in \mathcal{P}$.
Set $c(u) = \lambda$ for $u \in V - S$ and $c(v) = 1/M$ for $v \in \mathbf{A}$
for $r = 1$ to $\lfloor \log_{1+\epsilon} \frac{1+\epsilon}{\delta} \rfloor$ **do**
  for $j = 1$ to M **do**
    while there exists path $P \in \mathcal{P}_j$ such that $\ell(P) < \delta(1+\epsilon)^r$ **do**
      Let $c(P) = \min_{u \in P} c(u)$
      Let $d \geq 1$ be the smallest integer such that
      $\sum_{v \in P-S} \ell(v)\left(1 + \frac{\epsilon c(P)}{c(v)}\right)^d \geq \delta(1+\epsilon)^r$
      $z(P) \leftarrow z(P) + d \cdot c(P)$
      For $v \in P - S$, $\ell(v) \leftarrow \ell(v)\left(1 + \frac{\epsilon c(P)}{c(v)}\right)^d$
    end while
  end for
end for
Scale ℓ values such that $\ell(v) = \frac{\ell(v)}{\ell_{max}}$ where $\ell_{max} = \max_{u \in V \setminus S} \ell(u)$
Return ℓ

Algorithm 4 LSearch-Saa(M, B)

Set $\lambda = \frac{1}{MB}$
$\ell =$ MwuSaa(λ)
while $\sum_{u \in V \setminus S} \ell(u) > B$ **do**
  $\lambda = 2 * \lambda$
  $\ell =$ MwuSaa(λ)
end while
return ℓ

Algorithm 5 MwuRound(G, S, M, B, p, ϵ)

$\ell =$ LSearch-Saa(M, B)
Using the randomized rounding in [83], round the fractional solution ℓ to an integral solution X
$\mathbf{X} = \{u : u \in V \setminus S$ and $X(u) = 1\}$ is the set of nodes picked for intervention
return **X**

time are summarized in Lemmas 24 and 25. Then, we show that MwuRound, by following the same rounding approach as in saaRound, gives a bicriteria approximate solution to EpiControl problem by losing only a factor of $(1+4\epsilon)$.

Lemma 24. *Algorithm* MwuSaa *terminates after at most* $nM \log_{1+\epsilon} \frac{1+\epsilon}{\delta}$ *iterations.*

Proof. In each iteration, we increase the length of the minimum capacity node along path $P \in \mathcal{P}$ by a factor of $1+\epsilon$. For every node $u \in V \setminus S \cup \mathbf{A}$, $\ell(u) = \delta$ at the beginning of MwuSaa. The length of any variable at end of all iterations is at most $1+\epsilon$. The number of iterations in which any node is the minimum capacity node on the path chosen in an iteration is at most $\log_{1+\epsilon} \frac{1+\epsilon}{\delta}$ (as the length of each node starts with δ and can only be increased up to $1+\epsilon$). We say that a node is saturated when its length (given by the assignment) can no longer be increased. There are a total n nodes and at most nM target nodes (since each sampled graph can have up to n targets). Therefore, the number of iterations needed for all variables corresponds to all nodes to be saturated is $nM \ \log_{1+\epsilon} \frac{1+\epsilon}{\delta}$. □

Lemma 25. *Let* $\delta = (1+\epsilon)((1+\epsilon)L)^{-\frac{1}{\epsilon}}$. *Then, the total runtime of Algorithm* MwuSaa *is* $\tilde{O}(\frac{1}{\epsilon^2} nmM^2)$.

Proof. Let T_{sp} denote the time taken by an algorithm to find single-source shortest paths. Since there are exactly M samples, we need to use this algorithm M times per iteration r of MwuSaa. Using dijkstra's algorithm for shortest path computation, we have $T_{sp} = O(m)$. From Lemma 24, we know that MwuSaa needs at most $nM \ \log_{1+\epsilon} \frac{1+\epsilon}{\delta}$ iterations. Therefore, the total runtime is $O(nmM^2 \log_{1+\epsilon} \frac{1+\epsilon}{\delta})$. By setting $\delta = (1+\epsilon)((1+\epsilon)L)^{-\frac{1}{\epsilon}}$, where L could be $O(n)$ in the worst-case, the total runtime of Algorithm 1 is $\tilde{O}(\epsilon^{-2} nmM^2)$. □

Lemma 26. *Let* g_t *be the total flow computed by Algorithm* MwuSaa. *Then, there exists a feasible flow (i.e., a feasible solution to the dual) of value* $\frac{g_t}{\log_{1+\epsilon} \frac{1+\epsilon}{\delta}}$

Proof. Consider any node $u \in V_{H_j} \bigcup A(j)$ in H_j. For every $c(u)$ units of flow routed through u the value of $\ell(u)$ increases by a factor of at least $(1+\epsilon)$. Since, the algorithm stops when all paths have length > 1, on the last update to $\ell(u)$, the shortest path considered in that iteration must be of length < 1. Any increase in the value of $\ell(u)$ is at most $(1+\epsilon)$, therefore, at the end of the algorithm, $\ell(u) < 1+\epsilon$. We know that, initially, $\ell(u) = \delta$, therefore, the total flow through u is at most $c(u)\log_{(1+\epsilon)} \frac{(1+\epsilon)}{\delta}$. Therefore, by scaling the total flow g_t by a factor $\log_{(1+\epsilon)} \frac{(1+\epsilon)}{\delta}$ all the capacity constraints will be satisfied resulting in a feasible flow. □

Theorem 27. *Let $\mathbf{x}^*, \mathbf{y}^*$ denote an optimal solution to $LP_\ell(\lambda)$. Let ℓ be the solution returned by* MwuSaa(λ). *Then, $\mathbf{x}, \mathbf{y}$, defined as $x_u = \ell(u)$ for $u \in V$ and $y_v = \ell(v)$ for $v \in \bigcup_j A(j)$, is a feasible solution to $LP_\ell(\lambda)$, and $\sum_v c(v)\ell(v) \leq (1+4\epsilon)\mathrm{Z}_{LR}(\lambda)$.*

Proof. Algorithm MwuSaa is an adaptation of algorithm 2.2 of [30], which corresponds to $d = 1$ in the while loop. We show that the same proof holds here as well.

Let $\alpha(\ell) = \min_{P\in\mathcal{P}} \ell(P)$ denote the length of the shortest path with respect to ℓ, and let $D(\ell) = \sum_u c(u)\ell(u)$. Let $r_{max} = \lfloor \log_{1+\epsilon} \frac{1+\epsilon}{\delta} \rfloor$. Let $\beta^a = \min_{j=1}^{r_{max}} D(\ell_j)/\alpha(\ell_j)$ be the lowest objective value to $LP_\ell(\lambda)$ obtained by MwuSaa algorithm over all iterations. Note that β^a is a feasible by definition of α. Let $\ell_i(\cdot)$ denote the length function at the end of the ith iteration of the while loop, and let $\alpha(i) = \alpha(\ell_i)$, and $D(i) = D(\ell_i)$. For path P, let $z_i(P)$ denote the flow at the end of the ith iteration, and let $g_i = \sum_P z_i(P)$ denote the total flow at the end of the ith iteration.

As in [30], we now consider how $\alpha(i)$ changes over iterations. Let P denote

the path picked in iteration i. We have

$$\begin{aligned}
D(i) &= \sum_u \ell_i(u)c(u) \\
&= \sum_u \ell_{i-1}(u)c(u) + \sum_{u\in P} c(u)\ell_{i-1}(u)\Big[\Big(1+\frac{\epsilon c(P)}{c(u)}\Big)^d - 1\Big] \\
&= D(i-1) + \sum_{u\in P} c(u)\ell_{i-1}(u)\Big[\sum_{j=1}^{d}\Big(1+\frac{\epsilon c(P)}{c(u)}\Big)^j - \Big(1+\frac{\epsilon c(P)}{c(u)}\Big)^{j-1}\Big] \\
&= D(i-1) + \sum_{u\in P} c(u)\ell_{i-1}(u)\Big[\sum_{j=1}^{d}\Big(1+\frac{\epsilon c(P)}{c(u)}\Big)^{j-1}\Big(\Big(1+\frac{\epsilon c(P)}{c(u)}\Big) - 1\Big)\Big] \\
&= D(i-1) + \epsilon c(P)\sum_{u\in P} \ell_{i-1}(u)\Big[\sum_{j=1}^{d}\Big(1+\frac{\epsilon c(P)}{c(u)}\Big)^{j-1}\Big] \\
&= D(i-1) + \epsilon c(P)\sum_{j=1}^{d}\sum_{u\in P} \ell_{i-1}(u)\Big(1+\frac{\epsilon c(P)}{c(u)}\Big)^{j-1}
\end{aligned}$$

We use the fact that $\ell_{i-1}(u)\Big(1+\frac{\epsilon c(P)}{c(u)}\Big)^{j-1}$ is upper bounded by $(1+\epsilon)\alpha(i-1)$ to obtain

$$\begin{aligned}
&\leq D(i-1) + \epsilon c(P)\sum_{j=1}^{d}(1+\epsilon)\alpha(i-1) \\
&= D(i-1) + \epsilon d\cdot c(P)(1+\epsilon)\alpha(i-1) \\
&= D(i-1) + \epsilon(1+\epsilon)(g_i - g_{i-1})\alpha(i-1)
\end{aligned}$$

which, as in [30], gives

$$D(i) \leq D(0) + \epsilon(1+\epsilon)\sum_{j=1}^{i}(g_j - g_{j-1})\alpha(j-1) \tag{7.5}$$

Let γ denote the ratio of dual and primal solutions, i.e., $\gamma = \frac{\beta^a}{g_t}\log_{1+\epsilon}\frac{1+\epsilon}{\delta}$. This is because $g_t/\log_{1+\epsilon}\frac{1+\epsilon}{\delta}$ is a feasible dual solution. From the proof as in [30], we have the following upper bound on the $\frac{\beta^a}{g_t}$

$$\frac{\beta^a}{g_t} \leq \frac{\epsilon(1+\epsilon)}{\ln(\delta n)^{-1}}$$

For $\epsilon < .15$, substituting this bound in equation for γ, we obtain

$$\gamma = \frac{\beta^a}{g_t} \log_{1+\epsilon} \frac{1+\epsilon}{\delta} \leq 1 + 4\epsilon.$$

□

The theoretical guarantees of MwuRound are summarized in Theorem 28.

Theorem 28. *Let* $\mathbf{X}$ *denote the solution computed by Algorithm* MwuRound *for a given* $\epsilon > 0$. *If* $M = \Omega(n^2 \log n)$, *with probability at least* $1/2$, *we have* $\mathbb{E}[\#infections(\mathbf{X})] \leq 6(1+4\epsilon)\ \mathbb{E}[\#infections(\mathbf{X}^*)]$, *and* $|\mathbf{X}| \leq 12 \log(4nMN)B$, *where* $\mathbf{X}^*$ *denotes an optimal solution.*

Proof. From Theorem 27, we know that the fractional solution obtained by MwuSaa is $(1+4\epsilon)$-approximate solution for any λ. LSearch-Saa uses binary search to obtain a $(1+4\epsilon)$-approximate fractional solution for λ' that has $\sum_u x_u \leq B$. Since, we use the same rounding procedure from saaRound algorithm, from Theorem 11, we have $\mathbb{E}[\#\text{infections}(\mathbf{X})] \leq 6(1+4\epsilon)\ \mathbb{E}[\#\text{infections}(\mathbf{X}^*)]$ and $|\mathbf{X}| \leq 12 \log(4nMN)B$. □

7.2.2 Improving running time

Lemma 24 gives the same bound on the number of iterations as [30]. By exploiting the problem structure (i.e., we have a set of sampled subgraphs), and better data structures, the number of iterations can be made independent of M in the following manner; the total running time improves by a factor of M, compared with Lemma 25.

- Let $R(e)$ denote the set of samples containing edge e.
- Consider an iteration in which a path $P = s, v_1, \ldots, v_k \in \mathcal{P}_j$ is found, with $\ell(P) \leq \delta(1+\epsilon)^r$ and $v_k \in A(j)$

- We compute $R = \bigcap_{e \in P} R(e)$, the set of samples containing P.
- Perform updates for all the paths $P' = s, v_1, \ldots, v_{k-1}, a(v_{k-1}, j')$ for $j' \in R$.

The above modification to MwuSaa requires $O(n \log_{1+\epsilon} \frac{1+\epsilon}{\delta})$ shortest path computations, and has a total running time of $\tilde{O}(\epsilon^{-2} nmM)$. This is summarized in the lemma below.

Lemma 29. MwuSaa *can be implemented using* $O(n \log_{1+\epsilon} \frac{1+\epsilon}{\delta})$ *shortest path computations, and has a total running time of* $\tilde{O}(\epsilon^{-2} nmM)$.

7.3 Improving the scaling and memory usage of MwuSaa

Although MwuSaa scales to much larger networks than the LP solver, it is slow on large networks and is not memory efficient, as it needs to stores all the M sampled graphs in memory. The memory aspect can be handled by storing the sampled graphs in files, so only one sampled graph is loaded in the memory at any time. However, this solution does not improve the runtime. The main bottleneck in MwuSaa is that in each iteration, the algorithm has to iterate over all the M sampled graphs. Therefore, in this section, we present MwuScalable — a memory-efficient and scalable version of MwuSaa. In our experiments, we show that MwuScalable is able to scale to very large networks, corresponding to state- and level populations as well as has good performance guarantees in practice.

Main ideas in MwuScalable.

1. **Generate random stubs.** The intuition behind this approach is that the actual samples do not matter, as long as we are able to generate the

paths that would appear in these samples in each iteration of the algorithm. Let the probability that u is reachable from S in a sampled graph be denoted by $sp(u)$, and is referred to as stub probability. This can be estimated from our sampling process as follows: $sp(u) \approx \frac{reachable(u,S,M)}{M}$, where $reachable(u, S, M)$ denotes the number of samples in the M sampled graphs in which u is reachable from sources S. At the start of MwuScalable, we generate the random set of stubs $A_{sp}(u)$ for each node u as follows: for each $u \in V$ and $j \in 1, \cdots, M$, the stub $a(u, j)$ is generated with probability $sp(u)$. Let $\mathbf{A}_{sp} = \bigcup_u A_{sp}(u)$. The initial length $\ell(v) = \delta$ for each $v \in \mathbf{A}_{sp}$.

2. **Generate sampled graphs on the fly.** In every iteration r, we generate only $q \ll M$ sampled graphs, $H'_j = (V_{H'_j}, E_{H'_j})$ for $j \in [1, q]$. The algorithm then works on one sampled graph at a time. Therefore, at any time, the algorithm needs to store only one sampled graph in memory.

3. **Phases and iterations of the algorithm.** A phase of the algorithm corresponds to loop on line 4 in Algorithm 6, i.e., phase r corresponds to the r^{th} phase where the threshold is $threshold(r) = \delta(1 + \epsilon)^r$. In each phase r, the algorithm perform several iterations (depending on parameter q). Each iteration q of phase r, the algorithm generates a random sampled graph H'_j. Then, attaches a random stub for each node reachable from sources in H'_j to form H_j. Then, it iteratively updates lengths of paths in the sampled graph H_j until all paths are of length at least $threshold(r)$. Let $\mathcal{P}_j$ be set of paths from S to a stub node in H_j. The algorithm terminates after $r_{max} = \lfloor \log_{1+\epsilon} \frac{1+\epsilon}{\delta} \rfloor$ phases (same as MwuSaa).

4. **Computation of y values.** Some of the $\ell(u)$ for $u \in V$ could have a value in $[1, 1+\epsilon]$. Therefore, to make the solution $\mathbf{x}$ feasible, we scale $\ell(u) = \frac{\ell(u)}{\ell_{max}}$ where $\ell_{max} = max_{u \in V} \ell(u)$. Since the sampled graphs H_r generated in each iteration are a combination of paths from many sampled graphs, the $\ell(v)$

variables for $v \in \mathbf{A}_{sp}$ will not be meaningful. Therefore, we use the $\ell(u)$ for $u \in V$ obtained after scaling, and re-compute $\ell(v)$ (corresponding to y_{vj} variables) for $v = a(u, j)$ as follows: for each sampled graph H'_j, find a shortest path tree of H'_j with S as sources using $\ell(u)$ for $u \in V \setminus S$ as weights. For each $u \in V_{H'_j}$, let P_{uj} be a shortest path to node u that has length $\ell(P_{uj})$. Then, for the stub $v = a(u, j)$, we set $\ell(v) = 1 - \ell(P_{uj})$.

Algorithm 6 MwuScalable (λ)
Input: parameters λ and $q \in [1, M]$ (we assume the network $G = (V, E), S$, ϵ, and δ are fixed)
Output: ℓ

1: Generate set of random stubs $A_{sp}(u)$ for each $u \in V - S$ as follows: a stub $a(u, j)$ for $j \in \{1, \cdots, M\}$ is in $A(u)$ with a probability $sp(u)$. Let $\mathbf{A}_{sp} = \bigcup_u A_{sp}(u)$.
2: Initialize $\ell(u) = \delta$ for all $u \in (V - S) \bigcup \mathbf{A}_{sp}$, $\quad z(P) = 0$ for all $P \in \mathcal{P}$.
3: Set $c(u) = \lambda$ for $u \in V - S$ and $c(v) = \frac{1}{M}$ for $v \in \mathbf{A}_{sp}$.
4: **for** $r = 1$ to $\lfloor \log_{1+\epsilon} \frac{1+\epsilon}{\delta} \rfloor$ **do**
5: **for** $j = 1$ to q **do**
6: Generate a sampled graph $H'_j = (V_{H'_j}, E_{H'_j})$.
7: Construct $H_j = (V_{H_j}, E_{H_j})$ as follows: H_j is initially a copy of H'_j. For each $u \in V_{H'_j}$, pick a stub v_u uniformly at random from $A_{sp}(u)$. Then, $E_{H_j} = E_{H_j} \cup \{(v, v_u)\}$.
8: **while** there exists path $P \in \mathcal{P}_j$ such that $\ell(P) < \delta(1 + \epsilon)^r$ **do**
9: Let $c(P) = \min_{w \in P} c(w)$
10: Let $d \geq 1$ be the smallest integer such that $\sum_{w \in P - S} \ell(w) \left(1 + \frac{\epsilon c(P)}{c(w)}\right)^d \geq \delta(1 + \epsilon)^r$
11: $z(P) \leftarrow z(P) + d \cdot c(P)$
12: For each $v \in P - S$, $\ell(v) \leftarrow \ell(v) \left(1 + \frac{\epsilon c(P)}{c(v)}\right)^d$
13: **end while**
14: **end for**
15: **end for**
16: for each $v \in V - S$, $\quad \ell(v) = \frac{\ell(v)}{\ell_{max}}$ where $\ell_{max} = \max_{v \in V \setminus S} \ell(v)$
17: Recompute $\ell(v)$ for each $v = a(u, j)$ as follows: for each sampled graph H'_j, find a shortest path tree of H'_j using ℓ as weights. For each $u \in V_{H'_j}$, let P_{uj} be a shortest path of length $\ell(P_{uj})$. Then, $\ell(v) = 1 - \ell(P_{uj})$ for $v = a(u, j)$.
18: **return** ℓ

Lemma 30. *The solution ℓ computed by* MwuScalable *is a feasible solution to $LP_\ell(\lambda)$.*

Proof. First, $\ell(u)$ for each $u \in V \setminus S$ is scaled such that $\ell(u) \leq 1$. By the computation of $\ell(v)$ values in last step of the algorithm, we guarantee that the resultant ℓ gives a feasible solution. □

7.3.1 Modification for a set of budgets

We consider the version of EpiControl where a set of budgets $\mathcal{B} = \{B_1, \cdots, B_k\}$ are provided instead of a single budget B. Let us assume that the budgets in set $\mathcal{B}$ are sorted in non-increasing order. We observe that as λ increases, the fractional budget $(\sum_{u \in V} \ell(u))$ computed by MwuSaa (and MwuScalable) decreases. Therefore, for $B_i < B_j$ such that $B_i, B_j \in \mathbf{B}$, the values of λ consider by LSearch-Saa (and LSearch-Scalable) for B_i is a subset of λ values that are considered for B_j. Therefore, to avoid such duplicate computations, the starting value of λ for the budget B_i can be fixed as the λ value at termination for B_j.

7.3.2 Parallel approach

LSearch-Scalable sequentially searches over the λ values. But, we notice that this is an embarrassingly parallel task, as the computation for different λ values is independent of each other. Therefore, we can search over many λ values in parallel — this version of LSearch-Scalable is referred to as LSearch-Parallel. Further details on the implementation are provided in Section 7.4

7.4 Experiments

We addressed the following questions in our experiments:

1. **Performance.** What are the empirical guarantees of our methods? How does the performance of our approach compare to the baselines for this

problem? When to choose this algorithm instead of saaRound or vice-versa?

2. **Impact of parameters.** How do the runtime and solution quality of our methods vary with changes in transmission probability p and error parameter ϵ?

3. **Scaling**: How does the runtime of our approach grow with that of the size of the network? Does our approach scale to networks corresponding to state- and country-level populations?

4. **Parallelism**: What is the throughput of our parallel approach? How does the runtime vary with the number of threads used?

7.4.1 Datasets and Methods

In our experiments, we considered networks of different classes and varying sizes for the evaluation of the performance and scalability of our approach. Some of these networks are earlier described in Chapter 4. The random network (PA1) is based on the preferential attachment model [7] and the real-world collaboration networks, such as CA-GrQc and CA-HepTh [56]), is mainly used to evaluate the performance of our methods. We consider synthetic agent-based populations for Montgomery county in Virginia, Portland city in Oregon, and Virginia state, constructed based on first principles in [8,17,27]. These networks have been used in various public studies [88] as well as in works on intervention algorithms [83]. These networks also have demographic information, for each node in the network, such as age, income, location, etc. Finally, the networks Regional and US-size — which are generated using many copies (5 and 44 respectively) of the Virginia network, where the copies are connected by random edges — are mainly used for the scalability study. The datasets are summarized in Table 7.2.

Dataset	Nodes	Edges
Preferential1 (PA1)	1000	1996
CA-GrQc	5242	14496
CA-HepTh	9877	25998
Montgomery	75457	648667
Portland	2336693	8307767
Virginia	7605430	165533061
Regional	35024319	2068241728
US-size	334638920	32740251903

Table 7.2: Description of datasets

Methods and baselines. In our experiments, we consider the following methods listed below:

- LSearch-Saa. Obtains a feasible fractional solution ℓ using the subroutine MwuSaa.
- LSearch-Scalable. Obtains a feasible fractional solution ℓ using the subroutine MwuScalable.
- LSearch-Parallel. Obtains a feasible fractional solution ℓ using the subroutine MwuScalable for different λ values in parallel.
- MwuRound-Saa. Sequential version of MwuRound using LSearch-Saa.
- MwuRound-Scalable. Sequential version of MwuRound using LSearch-Scalable.
- MwuRound-Parallel. Parallel version of MwuRound using LSearch-Parallel.
- saaRound [83]. This algorithm obtains a fractional solution using the LP solver. Then rounds it to obtain an integral solution.
- Degree. The baseline that returns the set of B top-degree nodes in $V \setminus S$ as the intervention set.

- No-Action. Baseline in which no interventions are performed. The number of infections is given by #infections($\mathbf{X}$) for $\mathbf{X} = \emptyset$.

Performance measures. Below we describe the performance measures used in our experiments.

1. **Approximation ratio of a fractional solution ℓ.** We computed the approximation ratio of a fractional solution ℓ obtained by a method, for a given instance, as the ratio of the average number of infections $\frac{1}{M} \sum_j \sum_v y_{vj}$ resulting from ℓ to that of the LP_{saa} objective. So, this measure provides a comparison of this method with saaRound.

2. **Approximation ratio of the intervention set X.** We computed the approximation ratio of the integral solution $\mathbf{X}$ returned by a method, as the ratio of #infections($\mathbf{X}$) to the optimal objective value of LP_{saa}, which is a lower bound on the optimal objective value for the EpiControl instance. Therefore, this ratio is an upper bound on the approximation ratio of our methods.

3. **Budget violation of the intervention set X.** The budget violation (or the budget approximation ratio) of $\mathbf{X}$ returned by a method is the ratio of $|\mathbf{X}|$ to the given budget B.

In our experiments, we show the empirical performance guarantees of LSearch-Saa and LSearch-Scalable. Since, LSearch-Parallel runs the same subroutine MwuScalable as LSearch-Scalable does, the empirical guarantees shown for LSearch-Scalable also hold for this method.

Attack rate. The attack rate of an epidemic is the percentage of the population infected. We consider any attack rate in the range 10% to 20% as a moderate attack rate, whereas an attack rate $> 20\%$ is a high attack rate. On the other hand, an attack rate $< 10\%$ is seen as a low attack rate.

7.4.2 Performance.

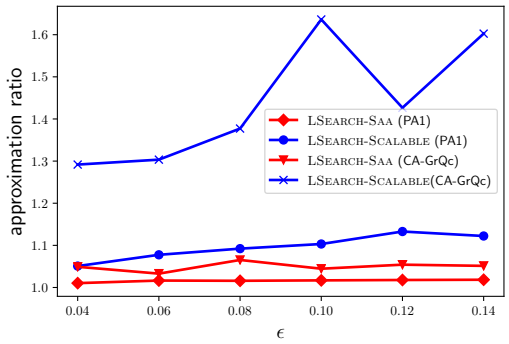


Figure 7.2: Comparison of approximation ratios of fractional solutions obtained by LSearch-Saa and LSearch-Scalable. The X-axis corresponds to the error parameter ϵ. $B = 50$.

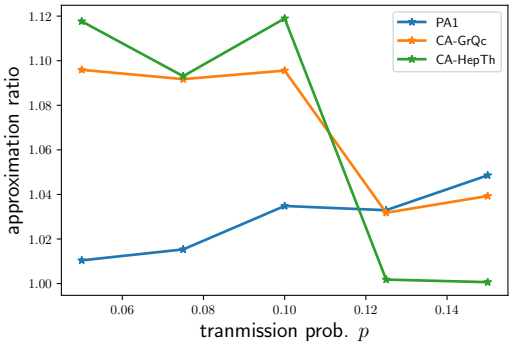


Figure 7.3: Impact of transmission probability p on approximation ratio of fractional solution obtained by LSearch-Scalable. The value of ϵ is set to 0.015.

Approximation ratio of fractional solutions. Figure 7.2 shows that the approximation ratio of the fractional solution ℓ obtained by LSearch-Saa is within 1.2 (i.e., its objective value is 1.1 $\times$ that of the optimal value of LP_{saa} objective), even for $\epsilon = 0.15$.

In comparison, the approximation ratio of the fractional solution obtained by LSearch-Scalable is at most 1.3 (Figure 7.2) for an epsilon value of 0.04. We note that the approximation ratio goes up to 1.7 for $\epsilon = 0.15$, which is within a factor of $(1 + 5\epsilon)$.

Figure 7.3 shows that LSearch-Scalable has significantly better performance for small values of ϵ. The performance of LSearch-Scalable is better on higher

p values on the collaboration networks. The approximation ratio of the fractional solution obtained by LSearch-Scalable is at most 1.12 $\times$ the optimal for all the p values and overall the networks considered in this experiment.

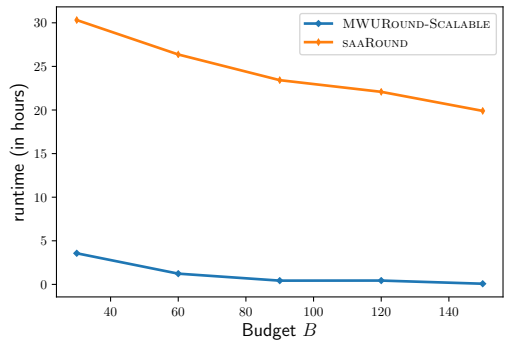


Figure 7.4: Montgomery. Runtime comparison: MwuRound-Scalable vs saaRound

7.4.3 Runtime performance.

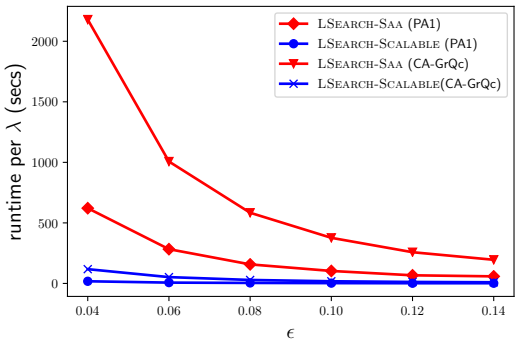


Figure 7.5: Runtime comparison of LSearch-Saa and LSearch-Scalable. The X-axis corresponds to the error parameter ϵ and the Y-axis corresponds to the runtime in seconds.

Figure 7.5 shows that for smaller values of ϵ, the runtime of LSearch-Scalable is about $\frac{1}{50}\times$ that of LSearch-Saa.

Figure 7.6 presents the runtime performance of LSearch-Scalable on various networks. The runtime reported here is for a single λ value and a moderate attack rate, except for the US-size network for which a high attack rate ($> 40\%$) is chosen. The total runtime taken by the algorithm on an instance of the dataset

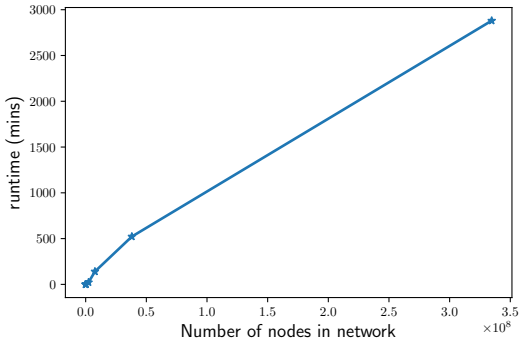


Figure 7.6: Runtime of LSearch-Scalable for a fixed λ and a medium attack rate (10-20% infections in population)

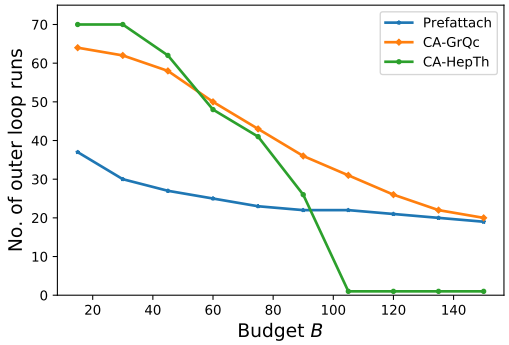


Figure 7.7: Number of outer loop iterations in LSearch-Scalable (i.e., no. of λ values needed to satisfy the budget constraint) for each budget B.

can be estimated by the run-time on any particular λ. The number of λ values considered by MwuRound-Scalable algorithm determines the runtime of both the sequential and the parallel versions of the algorithm. Figure 7.7 shows the depth of λ search for instances on different networks. As expected, the number of λ values considered decreases with the increase in the budget B. LSearch-Scalable ran within a few minutes ($<$ 15 minutes averaged over a few runs), for a fixed λ value, on the Portland network for problem instances with a moderate attack rate, whereas it ran about 2 hours and 9 hours on Virginia and Regional networks respectively. Finally, it ran in just about 2 days on the US-size network which has over 334 million nodes and 32 billion edges — for instance with a high attack rate.

7.4.4 Parallel Implementation

Figure 7.7 shows the relationship between the budget and the number of iterations of the outer loop (different λ values) needed for LSearch-Scalable to converge to a solution. One possible approach to speed up the convergence of the algorithm is to leverage parallel execution and to concurrently explore multiple λ values. To validate the efficiency of this idea, we have implemented a parallel version of the algorithm in C++ and OpenMP that leverages thread-level parallelism to explore multiple values of λ in batches and returns solution ℓ obtained for the largest λ value satisfying the budget constraint ($\sum_u \ell(u) \leq B$). This parallel implementation, and the corresponding experiments, are done by our collaborators.

We executed our implementation on a system equipped with 8 Intel© Xeon© Platinum 8276M CPU (28 cores per CPU, 224 cores total) running at 2.20GHz and $6TB$ of DRAM. In our experiments, we instructed the operating system to interleave the memory pages across all the 8 sockets in order to maximize the memory bandwidth available to the program. Our implementation uses a single shared copy of the graph in the compressed sparse row format (CSR) that requires $O(n + m)$ bytes to be stored. Each sample is stored through a graph view that is storing the active edges through a bitmap that requires $O(m)$ bits. $O(Mm)$ bits are required to store the entire collection of samples. The stub nodes are not stored directly in the graph, but the corresponding weights are stored in a separate memo on a need basis.

We have studied the strong scaling behavior of the algorithm and found that the scaling behavior is highly data-dependent from both the input graph and the input parameters of the algorithm. Figure 7.8 shows that our parallel implementation scales reasonably well (2.98× speedup) when going from 2 to 16 threads when $B = 400$. After 16 threads performance started to degrade. Our performance analysis led us to think that memory bandwidth becomes the bottleneck.

When $B = 500$, the algorithm shows no scaling because in this configuration the algorithm needs to explore a single λ value to converge. However, we want to note that there is no systematic way of knowing a priori how many values of λ will be needed. Therefore, evaluating the throughput of the algorithm gives better insight into the validity of the parallel approach.

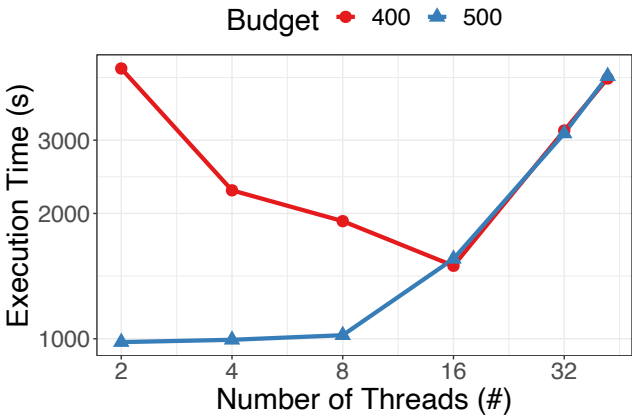


Figure 7.8: Strong scaling study on LSearch-Parallel for the Virginia network. Varying the budget shows the input dependent behavior of the algorithm.

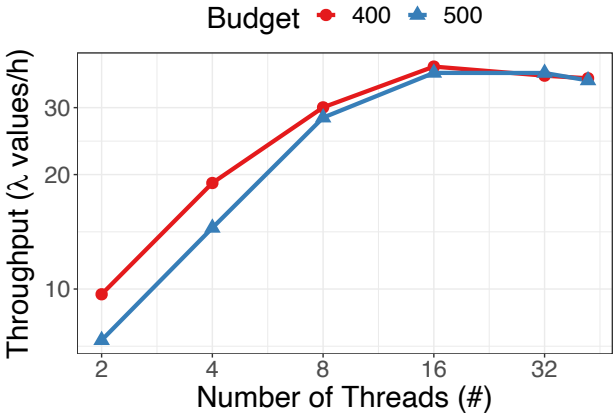


Figure 7.9: Number of λ values processed per hour by LSearch-Parallel on the Virginia network varing the budget.

Figure 7.9 shows how the throughput of LSearch-Parallel changes when increasing the number of threads and varying the budget on the Virginia network. We observed that, under our experimental settings, the peak in throughput is between 16 and 32 threads for our computing platform. When operating at

its maximum throughput, the MwuRound-Parallel algorithm shows to scale graciously with the size of the input network (Table 7.3).

Dataset	Time (s)
CA-HepTh	0.77
Montgomery	11.14
Portland	725.13
Virginia	1940.37

Table 7.3: Execution time of MwuRound-Parallel algorithm at the peak of throughput (16 threads). We report the execution time as the average of 3 consecutive runs.

Threads vs. execution time. The number of λ values to be considered, the number of threads used, and the memory needed for each thread together determine the ideal number of threads needed for each instance. We observe that using more threads reduces the execution time of MwuRound-Scalable for instances with smaller budgets.

7.5 Discussion and recommendation.

In our experiments, LSearch-Saa has approximation factors within a factor of $(1+2\epsilon)$, whereas LSearch-Scalable has approximation factors within $(1+8\epsilon)$ over all the networks considered.

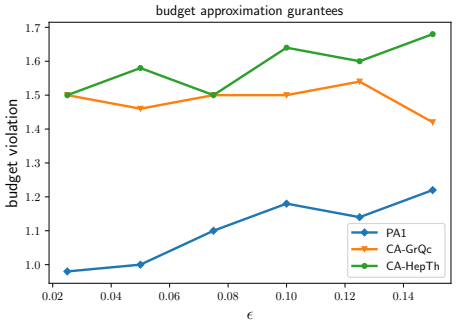


Figure 7.10: Budget violation of integral solution $\mathbf{X}$ obtained by MwuRound-SAA

The approximation ratios of the rounded solutions $\mathbf{X}$ obtained by both MwuRound-SAA and MwuRound-Scalable are close to 1 (as is the case for saaRound).

Figure 7.10 also shows that the budget violation of MwuRound-SAA is within 1.7 for ϵ as large as 0.15. There is a small upward trend in the budget violation as ϵ increases. The rounded solution $\mathbf{X}$ obtained by MwuRound-Scalable has similar performance guarantees as MwuRound, considering that both use the same rounding scheme.

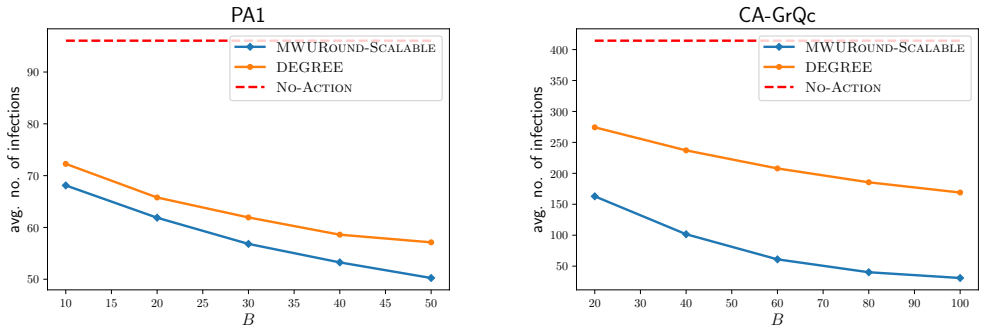


Figure 7.11: Comparison of MwuRound-Scalable with Degree and No-Action

Comparison to saaRound and Degree. Figure 7.11 shows that, MwuRound-Scalable outperforms the degree baseline. The #infections objective value of the Degree is at least 1.5 × that of the MwuRound-Scalable for CA-GrQc.

7.5.1 Runtime comparison.

Table 7.4 summarizes the runtime and space usage comparison of saaRound, MwuRound, and MwuRound-Scalable algorithms. MwuRound-Scalable is faster than the saaRound which uses the LP solver for networks with more than 10,000 nodes, such as Montgomery, as demonstrated in Figure 7.4. Our experiments demonstrate that the run-time shown for MwuRound-Scalable in this plot can be improved using MwuRound-Parallel. Both our methods, MwuRound-Scalable and MwuRound-Parallel, are able to scale well for larger networks than Portland such as Virginia, Regional, and even the US-size network which has over 334 million nodes and 32 billion edges.

Method	Runtime	Space
saaRound [83]	$O((n+nM)^{2.5})$	$O((n+nM)mM)$
MwuRound	$\tilde{O}(\epsilon^{-2}nmM)$	$O(nM+mM)$
MwuRound-Scalable	$\tilde{O}(\epsilon^{-2}nmq)$	$O(m+nM)$

Table 7.4: Runtime and space requirements of the different algorithms (see Table 7.1 for definitions of these quantities). We note that the space for MwuRound can be improved by a factor of M by using disk storage.

Recommendations.

- saaRound is a better choice for networks with fewer than 10000 nodes.
- Among the sequential methods, we recommend MwuRound-Scalable (with a small ϵ value) for large networks of the county- or city-scale populations such as Montgomery and Portland.
- MwuRound-Parallel is the obvious choice for very large networks corresponding to state- and country-level populations. The number of threads and the memory requirements determine the throughput of this approach as shown in our experiments.

Chapter 8

Conclusions

In this dissertation, our primary focus was on designing effective intervention strategies to control SIR class epidemics on networks. This is a challenging stochastic optimization problem. We developed approximation algorithms using stochastic optimization techniques for this problem. Our results showed that these techniques are quite effective in obtaining good approximation guarantees in practice. Our approach outperformed standard baselines for this problem.

However, we noticed that the use of LP solvers in our approach restricts its scalability. Therefore, we developed scalable algorithms that bypass the use of LP solver, by directly solving the LP, approximately. This was achieved by adapting the Multiplicative Weights Update (MWU) method for this problem. We showed that this improves the scalability of our approach to large networks corresponding to country-size populations.

Finally, we showed that our approach can be used to design effective group-scale interventions in the context of complex epidemic models (e.g. MultiPath) and other SIR class epidemic models. We also provided a framework to extend our approach to other epidemic models in SIR class dynamics.

Open Questions. Our work leads to several interesting open questions. Some of them are listed below:

(i) Is it possible to achieve tighter bounds on the performance guarantees of saaRound and MwuRound algorithms?

(ii) Can we study the characteristics of the nodes in near-optimal solutions in a statistically rigorous manner so as to identify "surrogates" for interventions?

(iii) Can we further improve the runtime of the scalable version of MwuRound so that it scales well to US population-scale networks?

Bibliography

[1] Rodolfo Acuna-Soto, David W Stahle, Matthew D Therrell, Sergio Gomez Chavez, and Malcolm K Cleaveland. Drought, epidemic disease, and the fall of classic period cultures in mesoamerica (ad 750–950). hemorrhagic fevers as a cause of massive population loss. *Medical hypotheses*, 65(2):405–409, 2005.

[2] R.M. Anderson and R.M. May. *Infectious Diseases of Humans*. Oxford University Press, Oxford, 1991.

[3] Elliot Anshelevich, Deeparnab Chakrabarty, Ameya Hate, and Chaitanya Swamy. Approximation algorithms for the firefighter problem: Cuts over time and submodularity. In Yingfei Dong, Ding-Zhu Du, and Oscar Ibarra, editors, *Algorithms and Computation*, pages 974–983, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[4] Sanjeev Arora, Elad Hazan, and Satyen Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory of Computing*, 8(1):121–164, 2012.

[5] Sanjeev Arora, Elad Hazan, and Satyen Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory of Computing*, 8(6):121–164, 2012.

[6] James Aspnes, Kevin Chang, and Aleksandr Yampolskiy. Inoculation strategies for victims of viruses and the sum-of-squares partition problem.

In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, SODA '05, pages 43–52, 2005.

[7] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

[8] Christopher L Barrett, Richard J Beckman, Maleq Khan, V. S. Anil Kumar, Madhav V Marathe, Paula E Stretz, Tridib Dutta, and Bryan Lewis. Generation and analysis of large synthetic social contact networks. In *Winter Simulation Conference*, pages 1003–1014. Winter Simulation Conference, 2009.

[9] Dimitris Bertsimas, Joshua Ivanhoe, Alexandre Jacquillat, Michael Li, Alessandro Previero, Omar Skali Lami, and Hamza Tazi Bouardi. Optimizing vaccine allocation to combat the covid-19 pandemic. *medRxiv*, 2020.

[10] Antonio Biondi, Raul Narciso C Guedes, Fang-Hao Wan, and Nicolas Desneux. Ecology, worldwide spread, and management of the invasive south american tomato pinworm, tuta absoluta: past, present, and future. *Annual Review of Entomology*, 63:239–258, 2018.

[11] John R. Birge and Franois Louveaux. *Introduction to Stochastic Programming*. Springer Publishing Company, Incorporated, 2nd edition, 2011.

[12] Bernoulli D. Blower S. An attempt at a new analysis of the mortality caused by smallpox and of the advantages of inoculation to prevent it. 1766. *Rev Med Virol.*, pages 275–88, 2004.

[13] Luisa C. C. Brant, Pedro C. Pinheiro, Isis E. Machado, Paulo R. L. Correa, Mayara R. Santos, Antonio L. P. Ribeiro, Unaí Tupinambás, Christine F. Santiago, Maria de Fatima M. Souza, Deborah C. Malta, and Valéria M. A.

Passos. The impact of COVID-19 pandemic course in the number and severity of hospitalizations for other natural causes in a large urban center in brazil. *PLOS Global Public Health*, 1(12):e0000054, December 2021.

[14] C. Castillo-Chavez, H. W. Hethcote, V. Andreasen, S. A. Levin, and W. M. Liu. Epidemiological models with age structure, proportionate mixing, and cross-immunity. *Journal of Mathematical Biology*, 27(3):233–258, May 1989.

[15] CDC. COVID-19 Scenario Modeling Hub, 2021.

[16] Parinya Chalermsook and Julia Chuzhoy. Resource minimization for fire containment. In *Proceedings of the Twenty-first Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '10, pages 1334–1349, Philadelphia, PA, USA, 2010. Society for Industrial and Applied Mathematics.

[17] Jiangzhuo Chen, Stefan Hoops, Achla Marathe, Henning Mortveit, Bryan Lewis, Srinivasan Venkatramanan, Arash Haddadan, Parantapa Bhattacharya, Abhijin Adiga, Anil Vullikanti, Mandy L Wilson, Gal Ehrlich, Maier Fenster, Stephen Eubank, Christopher Barrett, and Madhav Marathe. Prioritizing allocation of covid-19 vaccines based on social contacts increases vaccination effectiveness. *medRxiv*, 2021.

[18] Po-An Chen, Mary David, and David Kempe. Better vaccination strategies for better people. In *Proceedings of the 11th ACM Conference on Electronic Commerce*, EC '10, pages 179–188, New York, NY, USA, 2010. ACM.

[19] Herman Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *The Annals of Mathematical Statistics*, 23(4):493–507, December 1952.

[20] Reuven Cohen, Shlomo Havlin, and Daniel ben Avraham. Efficient immunization strategies for computer networks and populations. *Phys. Rev. Lett.*, 91:247901, Dec 2003.

[21] Catherine Constable, Nina R. Blank, and Arthur L. Caplan. Rising rates of vaccine exemptions: Problems with current policy and more promising remedies. *Vaccine*, 32(16):1793–1797, April 2014.

[22] Nedialko B Dimitrov and Lauren Ancel Meyers. Mathematical approaches to infectious disease prediction and control. In *Risk and Optimization in an Uncertain World*, pages 1–25. INFORMS, September 2010.

[23] Devdatt P. Dubhashi and Alessandro Panconesi. *Concentration of Measure for the Analysis of Randomized Algorithms.* Cambridge University Press, 2009.

[24] Jonathan Dushoff, Joshua B Plotkin, Cecile Viboud, Lone Simonsen, Mark Miller, Mark Loeb, and David J. D Earn. Vaccinating to protect a vulnerable subpopulation. *PLoS Medicine*, 4(5):e174, May 2007.

[25] Ken T.D. Eames, Jonathan M. Read, and W. John Edmunds. Epidemic prediction and control in weighted networks. *Epidemics*, 1(1):70–76, March 2009.

[26] David Eeasley and Jon Kleinberg. ECT volume 26 issue 5 cover and back matter. *Econ. Theory*, 26(5):b1–b4, October 2010.

[27] S. Eubank, H. Guclu, V. S. Anil Kumar, M. Marathe, A. Srinivasan, Z. Toroczkai, and N. Wang. Modelling disease outbreaks in realistic urban social networks. *Nature*, 429:180–184, 2004.

[28] S. Eubank, V. S.A Kumar, M. V Marathe, Aravind Srinivasan, and N. Wang. Structure of social contact networks and their impact on epidemics. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 70:181 – 181, 2006.

[29] Stephen Finbow and Gary MacGillivray. The firefighter problem: a survey of results, directions and questions. *Australasian J. Combinatorics*, 43:57–78, 2009.

[30] Lisa K. Fleischer. Approximating fractional multicommodity flow independent of the number of commodities. *SIAM J. Discret. Math.*, 13(4):505–520, October 2000.

[31] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[32] A. Ganesh, L. Massoulie, and D. Towsley. The effect of network topology on the spread of epidemics. In *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, volume 2, pages 1455–1466 vol. 2, 2005.

[33] C. Garcia-Martinez, C. Blum, F.J. Rodriguez, and M. Lozano. The firefighter problem. *Comput. Oper. Res.*, 60(C):55–66, aug 2015.

[34] Naveen Garg, Vijay V. Vazirani, and Mihalis Yannakakis. Multiway cuts in node weighted graphs. *J. Algorithms*, 50(1):49–61, January 2004.

[35] T.C. Germann, K. Kadau, I.M. Longini Jr, and C.A. Macken. Mitigation strategies for pandemic influenza in the united states. *Proceedings of the National Academy of Sciences*, 103(15):5935–5940, 2006.

[36] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer-Verlag, 1988.

[37] Reference manual for gurobi optimizer. `Gurobi.http://www.gurobi.com/`., 2017.

[38] Aric Hagberg, Pieter Swart, and Daniel S Chult. Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008.

[39] M. Elizabeth Halloran, Neil M. Ferguson, Stephen Eubank, Ira M. Longini, Derek A. T. Cummings, Bryan Lewis, Shufu Xu, Christophe Fraser, Anil Vullikanti, Timothy C. Germann, Diane Wagener, Richard Beckman, Kai Kadau, Chris Barrett, Catherine A. Macken, Donald S. Burke, and Philip Cooley. Modeling targeted layered containment of an influenza pandemic in the United States. In *Proceedings of the National Academy of Sciences (PNAS)*, pages 4639–4644, March 10 2008.

[40] Karin Hardt, Paolo Bonanni, Susan King, Jose Ignacio Santos, Mostafa El-Hodhod, Gregory D. Zimet, and Scott Preiss. Vaccine strategies: Optimising outcomes. *Vaccine*, 34(52):6691–6699, December 2016.

[41] Ara Hayrapetyan, David Kempe, Martin Pál, and Zoya Svitkina. Unbalanced graph cuts. In Gerth Stølting Brodal and Stefano Leonardi, editors, *Algorithms – ESA 2005*, pages 191–202, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[42] Xinran He and David Kempe. Robust influence maximization. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 885–894, New York, NY, USA, 2016. ACM.

[43] Herbert W Hethcote and Pauline van den Driessche. Two sis epidemiologic models with delays. *Journal of Mathematical Biology*, 40(1):3–26, 2000.

[44] Alexandra B Hogan, Peter Winskill, Oliver J Watson, Patrick G T Walker, Charles Whittaker, Marc Baguelin, Nicholas F Brazeau, Giovanni D Charles, Katy A M Gaythorpe, Arran Hamlet, Edward Knock, Daniel J

Laydon, John A Lees, Alessandra Løchen, Robert Verity, Lilith K Whittles, Farzana Muhib, Katharina Hauck, Neil M Ferguson, and Azra C Ghani. Within-country age-based prioritisation, global allocation, and public health impact of a vaccine against SARS-CoV-2: A mathematical modelling analysis. *Vaccine*, 39(22):2995–3006, May 2021.

[45] Oliver C. Ibe. Basic concepts in probability. In *Markov Processes for Stochastic Modeling*, pages 1–27. Elsevier, 2013.

[46] Thomas V. Inglesby. Public Health Measures and the Reproduction Number of SARS-CoV-2. *JAMA*, 323(21):2186–2187, 06 2020.

[47] David Isaacs. An ethical framework for public health immunisation programs. *New South Wales Public Health Bulletin*, 23(6):111, 2012.

[48] Matt J Keeling and Pejman Rohani. *Modeling infectious diseases in humans and animals*. Princeton university press, 2011.

[49] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proc. 9th KDD*, pages 137–146, 2003.

[50] William Ogilvy Kermack and Anderson G McKendrick. A contribution to the mathematical theory of epidemics. *Proceedings of the royal society of london. Series A, Containing papers of a mathematical and physical character*, 115(772):700–721, 1927.

[51] William Ogilvy Kermack and Anderson G McKendrick. Contributions to the mathematical theory of epidemics. ii.—the problem of endemicity. *Proceedings of the Royal Society of London. Series A, containing papers of a mathematical and physical character*, 138(834):55–83, 1932.

[52] William Ogilvy Kermack and Anderson G McKendrick. Contributions to the mathematical theory of epidemics. iii.—further studies of the problem

of endemicity. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 141(843):94–122, 1933.

[53] Jon Kleinberg. The small-world phenomenon: An algorithmic perspective. In *in Proceedings of the 32nd ACM Symposium on Theory of Computing*, pages 163–170, 2000.

[54] A. J Kleywegt, A. Shapiro, and T. Homem-de Mello. The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization*, 12(2):479–502, 2002.

[55] Tamara G. Kolda, Ali Pinar, Todd D. Plantenga, and C. Seshadhri. A scalable generative graph model with community structure. *CoRR*, abs/1302.6636, 2013.

[56] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graph evolution: Densification and shrinking diameters. *ACM Trans. Knowl. Discov. Data*, 1(1), March 2007.

[57] Eric Lofgren, M. Elizabeth Halloran, C. M. Rivers, J. M. Drake, Travis C. Porco, Bryan Lewis, Wan Yang, Alessandro Vespignani, Jeffrey Shaman, Joseph N. S. Eisenberg, M. C. Eisenberg, Madhav Marathe, Samuel V. Scarpino, Kathleen A. Alexander, Rafael Meza, Matthew J. Ferrari, James M. Hyman, Lauren A. Meyers, and Stephen Eubank. Opinion: Mathematical models: A key tool for outbreak response. *PNAS*, pages 18095–18096, 2014.

[58] Marjorie MacDonald, Bernadette Pauly, Geoff Wong, Kara Schick-Makaroff, Thea van Roode, Heather Wilson Strosher, Anita Kothari, Ruta Valaitis, Heather Manson, Warren O'Briain, Simon Carroll, Victoria Lee, Samantha Tong, Karen Dickenson Smith, and Megan Ward. Supporting

successful implementation of public health interventions: protocol for a realist synthesis. *Systematic Reviews*, 5(1), April 2016.

[59] M. Marathe and A. Vullikanti. Computational epidemiology. *Communications of the ACM*, 56(7):88–96, 2013.

[60] Ruth McCabe, Nora Schmit, Paula Christen, Josh C. D'Aeth, Alessandra Løchen, Dheeya Rizmie, Shevanthi Nayagam, Marisa Miraldo, Paul Aylin, Alex Bottle, Pablo N. Perez-Guzman, Azra C. Ghani, Neil M. Ferguson, Peter J. White, and Katharina Hauck. Adapting hospital capacity to meet changing demands during the COVID-19 pandemic. *BMC Medicine*, 18(1), October 2020.

[61] Joseph McNitt, Young Yun Chungbaek, Henning Mortveit, Madhav Marathe, Mateus R Campos, Nicolas Desneux, Thierry Brévault, Rangaswamy Muniappan, and Abhijin Adiga. Assessing the multi-pathway threat from an invasive agricultural pest: Tuta absoluta in asia. *Proceedings of the Royal Society B*, 286(1913):20191159, 2019.

[62] J. Medlock and A. P. Galvani. Optimizing influenza vaccine distribution. *Science*, 325(5948):1705–1708, 2009.

[63] Piet Van Mieghem, Dragan Stevanović, Fernando A. Kuipers, Cong Li, Ruud van de Bovenkamp, D. Liu, and Huijuan Wang. Decreasing the spectral radius of a graph by link removals. *Physical review. E, Statistical, nonlinear, and soft matter physics*, 84 1 Pt 2:016101, 2011.

[64] Ian F. Miller, Alexander D. Becker, Bryan T. Grenfell, and C. Jessica E. Metcalf. Disease and healthcare burden of COVID-19 in the united states. *Nature Medicine*, 26(8):1212–1217, June 2020.

[65] Joel C Miller and James M Hyman. Effective vaccination strategies for realistic social networks. *Physica A*, 386(2):780–785, December 2007.

[66] Marco Minutoli. *Parallel Influence Maximization Algorithms and Their Applications*. PhD thesis, Washington State University, 2021.

[67] Marco Minutoli, Prathyush Sambaturu, Mahantesh Halappanavar, Antonino Tumeo, Ananth Kalyananaraman, and Anil Vullikanti. Preempt: Scalable epidemic interventions using submodular optimization on multi-gpu systems. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–15, 2020.

[68] Michael Mitzenmacher and Eli Upfal. *Probability and Computing: Randomization and Probabilistic Techniques in Algorithms and Data Analysis*. Cambridge University Press, USA, 2nd edition, 2017.

[69] N E Moran, S Gainotti, and C Petrini. From compulsory to voluntary immunisation: Italy's national vaccination plan (2005-7) and the ethical and organisational challenges facing public health policy-makers across europe. *Journal of Medical Ethics*, 34(9):669–674, September 2008.

[70] John F Hernandez Nopsa, Gregory J Daglish, David W Hagstrum, John F Leslie, Thomas W Phillips, Caterina Scoglio, Sara Thomas-Sharma, Gimme H Walter, and Karen A Garrett. Ecological networks in stored grain: Key postharvest nodes for emerging pests, pathogens, and mycotoxins. *BioScience*, page biv122, 2015.

[71] Masaki Ogura and Victor M. Preciado. *Optimal Containment of Epidemics in Temporal and Adaptive Networks*, pages 241–266. Springer Singapore, Singapore, 2017.

[72] World Health Organization. *World health statistics 2019: monitoring health for the SDGs, sustainable development goals*. World Health Organization, 2019.

[73] Romualdo Pastor-Satorras, Claudio Castellano, Piet Van Mieghem, and Alessandro Vespignani. Epidemic processes in complex networks. *Reviews of Modern Physics*, 87(3):925–979, August 2015.

[74] Boivin G. Piret J. Pandemics throughout history. *Front Microbiol.*, 2021.

[75] S.A. Plotkin, D.B. Shmoys, and E. Tardos. Fast approximation algorithms for fractional packing and covering problems. In *[1991] Proceedings 32nd Annual Symposium of Foundations of Computer Science*, pages 495–504, 1991.

[76] B Aditya Prakash, Deepayan Chakrabarti, Michalis Faloutsos, Nicholas Valler, and Christos Faloutsos. Threshold conditions for arbitrary cascade models on arbitrary networks. In *2011 IEEE 11th International Conference on Data Mining*. IEEE, December 2011.

[77] Victor M. Preciado, Michael Zargham, Chinwendu Enyioha, Ali Jadbabaie, and George J. Pappas. Optimal vaccine allocation to control epidemic outbreaks in arbitrary networks. In *IEEE Conference on Decision and Control*. IEEE, 2013.

[78] Victor M. Preciado, Michael Zargham, Chinwendu Enyioha, Ali Jadbabaie, and George J. Pappas. Optimal resource allocation for network protection against spreading processes. In *IEEE Transactions on Control of Network Systems*, pages 99 – 108. IEEE, 2014.

[79] Victor M. Preciado, Michael Zargham, and David Sun. A convex framework to control spreading processes in directed networks. In *Annual Conference on Information Sciences and Systems (CISS)*. IEEE, 2014.

[80] Benjamin Ridenhour, Jessica M Kowalik, and David K. Shay. Unraveling r0: Considerations for public health applications. *American Journal of Public Health*, 108:S445–S454, 2015.

[81] Sudip Saha, Abhijin Adiga, B. Aditya Prakash, and Anil Vullikanti. Approximation algorithms for reducing the spectral radius to control epidemic spread. In *SIAM SDM*, 2015.

[82] Marcel Salathe and James H. Jones. Dynamics and control of diseases in networks with community structure. *PLoS Computational Biology*, 2012.

[83] Prathyush Sambaturu, Bijaya Adhikari, B. Aditya Prakash, Srinivasan Venkatramanan, and Anil Vullikanti. Designing effective and practical interventions to contain epidemics. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '20, page 1187–1195, Richland, SC, 2020. International Foundation for Autonomous Agents and Multiagent Systems.

[84] Prathyush Sambaturu and Anil Vullikanti. Designing robust interventions to control epidemic outbreaks. In Hocine Cherifi, Sabrina Gaito, José Fernendo Mendes, Esteban Moro, and Luis Mateus Rocha, editors, *Complex Networks and Their Applications VIII*, pages 469–480, Cham, 2020. Springer International Publishing.

[85] Alexander Shapiro. Monte carlo sampling methods. In *Handbooks in Operations Research and Management Science*, pages 353–425. Elsevier, 2003.

[86] Alexander Shapiro, Darinka Dentcheva, and Andrzej Ruszczyński. *Lectures on Stochastic Programming*. Society for Industrial and Applied Mathematics, January 2009.

[87] Alexander Shapiro and Anton Kleywegt. Minimax analysis of stochastic problems. *Optimization Methods and Software*, 17(3):523–542, 2002.

[88] Meghendra Singh, Prasenjit Sarkhel, Gloria J. Kang, Achla Marathe, Kevin Boyle, Pamela Murray-Tuite, Kaja M. Abbas, and Samarth Swarup. Impact

of demographic disparities in social distancing and vaccination on influenza epidemics in urban and rural regions of the united states. *BMC Infectious Diseases*, 19, 12 2019.

[89] David L Smith, Katherine E Battle, Simon I Hay, Christopher M Barker, Thomas W Scott, and F Ellis McKenzie. Ross, macdonald, and a theory for the dynamics and control of mosquito-transmitted pathogens. *PLoS pathogens*, 8(4):e1002588, 2012.

[90] Manisha Sudhir. Controlling diffusion on multi-pathway spatial networks: Application to biological invasions. Master's thesis, University of Virginia., 2021.

[91] Chaitanya Swamy and David B. Shmoys. Approximation algorithms for 2-stage stochastic optimization problems. *SIGACT News*, 37(1):33–46, March 2006.

[92] Guy Tennenholtz, Constantine Caramanis, and Shie Mannor. The stochastic firefighter problem. *CoRR*, abs/1711.08237, 2017.

[93] H. Tong, B. Aditya Prakash, T. Eliassi-Rad, M. Faloutsos, and C. Faloutsos. Gelling, and melting, large graphs by edge manipulation. In *CIKM*, 2012.

[94] S. Truelove, C. P. Smith, et al. Projected resurgence of covid-19 in the united states in july—december 2021 resulting from the increased transmissibility of the delta variant and faltering vaccination. *medRxiv*, 2021.

[95] S. Venkatramanan, J. Chen, S. Gupta, B. Lewis, M. Marathe, H. Mortveit, and A. Vullikanti. Spatio-temporal optimization of seasonal vaccination using a metapopulation model of influenza. In *2017 IEEE International Conference on Healthcare Informatics (ICHI)*, pages 134–143, Aug 2017.

[96] Veenapani Rajeev Verma, Anuraag Saini, Sumirtha Gandhi, Umakant Dash, and Shaffi Fazaludeen Koya. Capacity-need gap in hospital resources for varying mitigation and containment strategies in india in the face of COVID-19 pandemic. *Infectious Disease Modelling*, 5:608–621, 2020.

[97] Yang Wang, Deepayan Chakrabarti, Chenxi Wang, and Christos Faloutsos. Epidemic spreading in real networks: An eigenvalue viewpoint. In *In SRDS*, pages 25–34, 2003.

[98] Bryan Wilder, Sze-Chuan Suen, and Milind Tambe. Preventing infectious disease in dynamic populations under uncertainty. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18. AAAI Press, 2018.

[99] David P. Williamson and David B. Shmoys. *The Design of Approximation Algorithms*. Cambridge University Press, 2011.

[100] Dirk Witteveen and Eva Velthorst. Economic hardship and mental health complaints during covid-19. *Proceedings of the National Academy of Sciences*, 117(44):27277–27284, 2020.

[101] Yingrui Yang, Ashley McKhann, Sixing Chen, Guy Harling, and Jukka-Pekka Onnela. Efficient vaccination strategies for epidemic control using network information. *Epidemics*, 27:115–122, June 2019.

[102] Peng Zhang, Jin-Yi Cai, Lin-Qing Tang, and Wen-Bo Zhao. Approximation and hardness results for label cut and related problems. *Journal of Combinatorial Optimization*, 2009.

[103] Yao Zhang, Abhijin Adiga, Anil Vullikanti, and B Aditya Prakash. Controlling propagation at group scale on networks. In *Data Mining (ICDM), 2015 IEEE International Conference on*, pages 619–628. IEEE, 2015.

[104] Yao Zhang and B. Aditya Prakash. Dava: Distributing vaccines over networks under prior information. In *Proceedings of the SIAM Data Mining Conference*, SDM '14, 2014.