Sequential Network Planning Problems for Public Health Applications

a dissertation presented
by
Han-Ching Ou
to
The Department of Computer Science

in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy
in the subject of
Computer Science

Harvard University
Cambridge, Massachusetts
May 2022

Thesis advisor: Professor Milind Tambe Han-Ching Ou

Sequential Network Planning Problems for Public Health Applications

Abstract

In the past decade, breakthroughs of Artificial Intelligence (AI) in its multiple sub-area have made new applications in various domains possible. One typical yet essential example is the public health domain. There are many challenges for humans in our never-ending battle with diseases. Among them, problems involving harnessing data with network structures and future planning, such as disease control or resource allocation, demand effective solutions significantly. However, unfortunately, some of them are too complicated or unscalable for humans to solve optimally. This thesis tackles these challenging sequential network planning problems for the public health domain by advancing the state-of-the-art to a new level of effectiveness.

In particular, My thesis provides three main contributions to overcome the emerging challenges when applying sequential network planning problems in the public health domain, namely (1) a novel sequential network-based screening/contact tracing framework under uncertainty, (2) a novel sequential network-based mobile interventions framework, (3) theoretical analysis, algorithmic solutions and empirical experiments that shows superior performance compared to previous approaches both theoretically and empirically.

More concretely, the first part of this thesis studies the active screening problem as an emerging application for disease prevention. I introduce a new approach to modeling multi-round network-based screening/contact tracing under uncertainty. Based on the well-known network SIS model in computational epidemiology, which is applicable for many diseases, I propose a model of the multi-agent active screening problem (ACTS) and prove its NP-hardness. I further proposed the REMEDY (REcurrent screening Multi-round Efficient DYnamic agent) algorithm for solving this problem. With a time and solution quality trade-off, REMEDY has two variants, Full- and Fast-REMEDY. It is a Frank-Wolfe-style gradient descent algorithm realized by compacting the representation of belief states to represent uncertainty. As shown in the experiment conducted, Full- and Fast-REMEDY are not only being superior in controlling diseases to all the previous approaches; they are also robust to varying levels of missing information in the social graph and budget change, thus enabling the use of our agent to improve the current practice of real-world screening contexts.

The second part of this thesis focuses on the scalability issue for the time horizon for the ACTS problem. Although Full-REMEDY provides excellent solution qualities, it fails to scale to large time horizons while fully considering the future effect of current interventions. Thus, I proposed a novel reinforcement learning (RL) approach based on Deep Q-Networks (DQN). Due to the nature of the ACTS problem, several challenges that the traditional RL can not handle have emerged, including (1) the combinatorial nature of the problem, (2) the need for sequential planning, and (3) the uncertainties in the infectiousness states of the population. I design several innovative adaptations in my RL approach to address the above challenges. I will introduce why and how these adaptations are made in this part.

For the third part, I introduce a novel sequential network-based mobile interventions framework. It is a restless multi-armed bandits (RMABs) with network pulling effects. In the proposed model, arms are partially recharging and connected through a graph. Pulling one arm also improves the state of neighboring arms, significantly extending the previously studied setting of fully recharging bandits with no network effects. Such network effect may arise due to regular population movements (such as commuting between home and work) for mobile intervention applications. In my thesis, I show that network effects in RMABs induce strong reward coupling that is not accounted for by existing solution methods. I also propose a new solution approach for the networked RMABs by exploiting concavity properties that arise under natural assumptions on the structure of intervention effects. In addition, I show the optimality of such a method in idealized settings and demonstrate that it empirically outperforms state-of-the-art baselines.

Contents

Listing of figures

Acknowledgments

One of the best decisions I have made is to join my advisor, Milind Tambe, and work with his fantastic group. Thank you, Milind, for being incredibly patient and supportive throughout these years. It is extremely lucky for me to have the chance to learn and grow through your guidance. Your passion for research, changing the world, and making a real impact has profoundly influenced many people, including myself. I will always be grateful on my way forward.

I'm also very grateful to many coauthors that I have the honor to collaborate with: Bryan Wilder, Kayla de la Haye, Bistra Dilkina, Phebe Vayanos, Arunesh Sinha, Sze-Chuan Suen, Andrew Perrault, Alpan Raval, Marie Charpignon, Jackson Killian, Aditya Mate, Shahin Jabbari, Angel Desai, Maimuna Majumder, Haipeng Chen, Wei Qiu, Bo An, Christoph Siebenbrunner, Meredith B Brooks, David Kempe, and Yevgeniy Vorobeychik. My Ph.D. journey would not be the same without all the support I received from you. In addition, I would like to express my gratitude to the rest of my committee: Ariel Procaccia and Susan Murphy. Thank you for your insightful comments on this work which assist me in further polishing it.

I would like to extend my sincere thanks to all the wonderful people I met in teamcore at both Harvard and USC: Christoph Siebenbrunner, Andrew Perrault, Shahin Jabbari, Haipeng Chen, Arpita Biswas, Haifeng Xu, Kumar Amulya Yadav, Sara Marie McCarthy, Aaron Schlenker, Andrew Perrault, Bryan Wilder, Shahin Jabbari, Shahrzad Gholami, Bryan Wilder, Elizabeth Bondi, Aida Rahmattalabi, Kai Wang, Aditya Mate, Jackson Killian, Lily Xu, Sanket Shah, Paula Rodriguez Diaz, Sonja Johnson-Yu. All of you have supported me in different ways, and I really enjoyed our activities during the past few years. Special thanks to Kai Wang for being great as a labmate, roommate, and baker. In addition, to my close friends since college: Po-Jung Chang, Jen-Shuo Liu, Peng-Jui Wang, and Tzu-Yi Peng, thank you for being my friends for more than ten years through high and lows. Although we are scattered around the world now, neither time nor space seems to affect our friendship in the slightest way.

Finally, I want to thank my parents, Jo-Chiu Hsiao and Wei-Hsiung Ou. I could not go so far without their unconditional love,encouragement and support. I thank them for being so proud of even the smallest accomplishment I have. Their love and care have guided me through times of difficulties. I wish my Ph.D. degree can bring some joy to them in return.

1

Introduction

The science and art of preventing disease and promoting people's health has gained increasing attention as technologies advanced. Among these technologies, the development of various sub-areas in AI has made promising progress by reducing human physical and mental efforts to perform difficult and complex tasks Murphy et al. [2007], Dickerson et al. [2012]. One of the most complex yet widely applicable types of tasks is sequential planning on network structure. It requires dealing with a large number of combinatorial actions and future planning simultaneously. In the previous approach, health workers often struggled when planning for complex tasks. They either relied on naive guides that were short-sighted or nothing but their own experience. By studying sequential planning problems on network structure, we reveal the insight hidden behind the veil of these complex problems and help health workers make wiser decisions and plan more effectively.

1.1 Why Sequential Network Planning

Network analysis is a vital tool for public health analysis due to its flexibility and accuracy compared to other models. In addition, it facilitates communication between technical experts and stakeholders by providing a language to visualize and understand complex concepts. Furthermore, network models are usually amenable to scaleable computational techniques thanks to the numerous efforts to develop related theories and algorithms.

One example of network analysis in the public health domain is to model disease transmission, especially for infectious diseases such as tuberculosis, influenza, and sexually transmitted diseases (STDs) (e.g., gonorrhea and chlamydia). These contagious illnesses are responsible for millions of deaths every year. The ability to accurately model them enables further planning of inoculation or isolation and may significantly affect the mortality rate of a particular epidemic. In particular, network models are often more intuitive and accurate for predicting disease spread through heterogeneous host populationsBansal et al. [2007], Danon et al. [2011]. This is because the contact between individuals allows an infectious disease to propagate naturally define a network. Such a network will enable us to distinguish spreading events of different properties and gain insights into the epidemiological dynamics. The network epidemic model also opens up the possibility of active screening optimization (or contact tracing). Unlike homogeneous models, the transmission routes defined by the network made it possible to exploit the knowledge of structure for disease control. In my thesis, the sequential planning further captures the dynamic of disease spread compared to previous one-shot vaccination models. Therefore, it is clear the sequential planning on epidemic networks for active screening has a vast potential for improving the current practice.

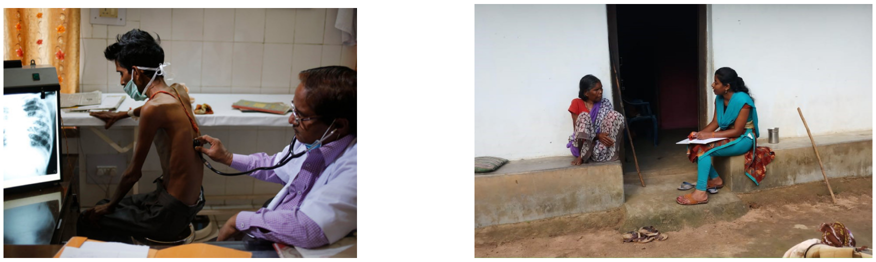
Another example of the network model is the commuting network of the potential patients' communities. Besides contact between individuals, their commuting behavior between different locations also naturally forms a network. Studying such a network allows the health service providers to engage (usually done by using mobile health clinics) the potential patients more effectively, allowing vulnerable communities to access otherwise not viable assistance Stephanie et al. [2017]. Again, the dynamic properties of the commuting behavior have made the sequential planning for the schedule difficult but essential, especially when the interplay between time and space are involved. The following section will elaborate on more background, challenges, and contributions to addressing the sequential network planning problems for these two example applications in this thesis.

1.2 Sequential Network Planning Problems Addressed

My thesis provides two representative sequential network planning problems with public health applications and their solutions.

1.2.1 Active Screening

Contagious diseases are critical public-health challenges that continue to threaten lives and impose significant economic burdens on society. For example, the economic loss due to influenza in the USA alone is estimated to be $11.2 billion in 2015 Putri et al. [2018]. While low-cost treatment programs are available, individuals ignore symptoms and delay care, increasing transmission risk. As a result, health agencies engage in active screening or contact tracing efforts as figure 1.1 shows, where individuals are asked to undergo diagnostic tests and offered treatment if tests are positive Eames & Keeling [2003], Cadman et al. [1984]. However, active screening is expensive in developing countries. Even in the USA, Braxton et al. Braxton et al. [2017] state that "In 2012, 52% of state and local STD programs experienced budget cuts. This amounts to reductions in clinic hours, contact tracing, and screening for common STDs." In India, an estimated 1 million missing tuberculosis (TB) cases require an efficient method of active screening, particularly given limited health budgets Chinnakali et al. [2016]. Efficiently identifying and intervening for infectious cases is therefore of vital importance.



(a) Passive screening

(b) Active screening

Figure 1.1: Passive screening only treats patients who come to a clinic voluntarily whereas active screening can treat patients in hard-to-reach tribal areas at a higher cost Tuberculosis & Disease [2018]

Active screening (or contact tracing) aims at selecting a subset of nodes in a social network for screening, so as to prevent the spread of transmissive diseases. When an individual is tested positive, they are marked as infected. The health worker, or contact tracer, records who else has been exposed and marks them as contacts or potentially infected individuals. The potentially infected individuals might not voluntarily seek treatment and testing. In case any of such individuals are infected, they can spread the disease further. Active screening aims to target these individuals and slow down the spread of the disease Eames & Keeling [2003], Taylor-Robinson [1994].

There is a huge body of literature on spread and control of recurrent diseases (no permanent immunity) Ball et al. [2015], Sun & Hsieh [2010], Wang [2005], Zhang & Prakash [2015], Ganesh et al. [2005]. However, these prior studies assume perfect observation of who is infected and who is not. Also, most of these methods focus on eradication of disease, which is not possible if the screening resources are limited. Thus, important real world characteristics such as partial observation and limited resources have not been adequately handled in prior work.

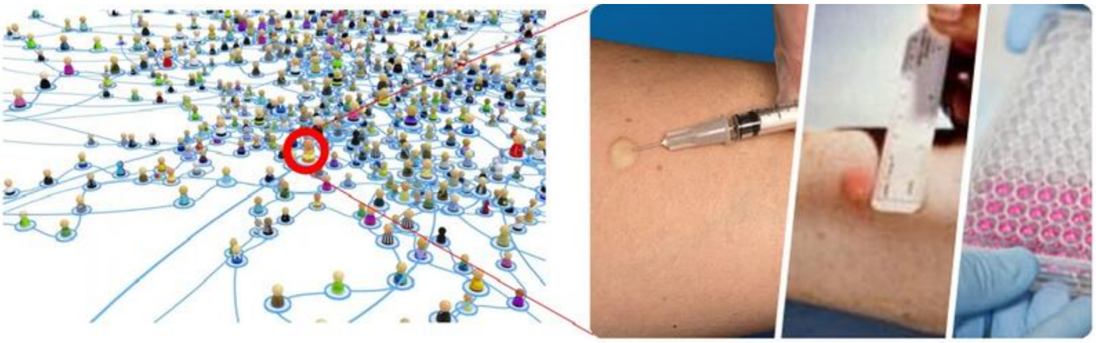


Figure 1.2: The active screening problem: Given as input a network of individuals, their contact, and disease transition model, the active screening problem is to provide a policy that maps the observations of the network to state to a set of individuals to screen every time step.

There are many challenges in implementing active screening. First, not every individual in a social network can be screened due to limited resources (in this case contact tracers or amount of tests available). Therefore, for each screening round, we need to optimally select a subset of nodes which is a *combinatorial optimization* problem. Second, the health states of the many individuals in the network are unknown (sans people who get tested actively or passively). Finally, the above challenges are significantly amplified when *sequential planning* is involved as we need to account for the future effects of current screening actions.

Active Screening-Contributions

To address this shortcoming in active screening of recurrent diseases, this thesis develop a model of the active screening problem and present different approaches for two different scenarios: short-term and long-term planning.

Model of active screening problem (ACTS): The *first contribution* for this thesis in active screening is a model of the multiagent active screening problem (ACTS). Given as input a network ($G(V, E)$) of individuals(V), their contact (E), and disease transition model, the active screening problem is to provide a policy that maps the observations of the network to state to a set of individuals to screen every time step. We focus on spread of recurrent infectious diseases modeled using the well-known network SIS model in computational epidemiology Wang et al. [2003], which is applicable for many diseases such as syphilis and typhoid. It is the foundation of more complex models that capture more disease dynamics (such as latent states, variation in birth/death rates, or multiple treatment states). The network SIS model is specified by a graph where nodes are individuals and edges indicate physical contact through which disease spread is probabilistic. ACTS models multi-agent interactions in that the nodes in the graph model individuals who interact with other individuals. The individuals can be either susceptible (S) or infected (I). The contribution of multiagent systems in computational epidemiology is well recognized in previous literature Swarup et al. [2014]. Our model further includes real-world constraints, namely that health workers are uncertain about the health state of individuals, have a small screening budget relative to the population size, and must engage in active screening over multiple rounds (time periods) due to recurrent of the disease. As a first result, we prove that the ACTS problem is NP-hard. To the best of our knowledge, no other model in the AI literature has considered multi-round active screening with partially observable health state for controlling disease spread.

Short-term Planning: For diseases of slow treatment and limitted horizon, I proposed an adaptive software agent, REMEDY (**RE**current screening **M**ulti-round **E**fficient **DY**namic agent) to address such scenario. The model is developed in cooperation with a research institute in India (name withheld for anonymity of authors), which partners with the Central Tuberculosis Division (CTD) of India to facilitate active screening for TB. REMEDY assists maximizing effectiveness of active screening under real world budgetary constraints and limited contact information. Such screening of TB patients currently takes place quarterly in over 50 districts scattered across 18 states of India. REMEDY is intended to assist health workers in India in their work in active screening in the field. Our software agent is currently under review before deployment as a means to improve the efficiency of district-wise active screening for tuberculosis in India, although REMEDY has applicability for active screening of other recurrent diseases.

The next contribution of this thesis toward active screening problem is two novel algorithms for short-term planning, FULL- and FAST-REMEDY. In the former, we consider the effect of both current and future screening actions to solve the ACTS problem. FULL-REMEDY achieves scale-up via an innovative combination of : (i) easier to optimize upper-bound of the ACTS objective; (ii) a Frank-Wolfe Style gradient descent algorithm; (iii) compact representation of belief states to represent uncertainty. FAST-REMEDY works in a similar fashion as FULL-REMEDY, but by optimizing just the current step actions runs almost two orders of magnitude faster than FULL-REMEDY in practice. As another contribution, we illustrate the benefits of FULL- and FAST-REMEDY via extensive testing on seven different real-world human contact networks against various baselines across a range of realistic disease parameters. For the largest network of ~76,000 individuals we see improvements in performance of almost 40% over the prior best method which directly maps to thousands of fewer infections every six months.

REMEDY is developed to assist screening for infectious diseases under conditions where screening tests are slow and expensive, budgets are limited, and information on the underlying social graph is available. As we also show, the performance improvements exhibited by FULL- and FAST-REMEDY are robust to varying levels of missing information in the social graph and budget change, thus enabling the use of our agent to improve the current practice of real-world screening contexts.

Long-term Planning: Due to the superior performance of RL approaches in solving long term planning problems Mnih et al. [2015], Silver et al. [2016, 2017], in my thesis I propose a novel RL approach that builds upon a powerful variant of RL called DQN Mnih et al. [2013]. We first formulate the multi-round active screening problem as a Markov Decision Process (MDP), where the state is a vector representing the probability of each node in the network being infected, and the action is to select which subset of nodes to actively screen. Due to the extremely high-dimensional state and action spaces, vanilla DQN algorithms cannot be directly applied to solve our problem efficiently. We therefore design several innovative adaptations over vanilla DQN that fully exploit the problem structure of multi-round active screening. First, we show that the node features in the underlying contact network are inter-correlated. To efficiently capture the intrinsic correlations between different nodes, we use GCNs as the function approximator to represent the Q-function. Second, because in each time period we need to select a subset of nodes to actively screen, this leaves vanilla DQN un-scalable as it needs to solve a combinatorial optimization problem in the action selection procedure. To avoid this we decompose the node set selection problem in each time period as a sub-sequence of decisions, and then design a novel two-level RL framework that solves the problem in a hierarchical manner. It has two types of agents. The primary agent works at the main sequence level and interacts with the environment, while multiple secondary agents work at the sub-sequence level and are responsible for generating actions sequentially within each time period. Last, we find that the reward signals for the secondary agents are sparse. To speed up the slow convergence of secondary agents' policies that arises from the sparseness of rewards, we incorporate ideas from curriculum learning into our algorithm. Intuitively, the algorithm warm-starts at the beginning of training with a simpler task, which has limited action choice and true state information. As the training goes on, the algorithm gradually increases task difficulty by providing uncertain state information and more action choice until the problem becomes the same as the original active screening problem.

The main contributions of this thesis for long-term planning active screening are summarized as follows. (i) I formulate the multi-round active screening problem for recurrent diseases as a Markov Decision Process (MDP). (ii) To solve the formulated MDP, I propose a novel solution algorithm on the basis of DQN, with several innovative adaptations that fully exploit the problem structure of the formulated MDP. (iii) Extensive experiments were conducted on various real-world networks with distinct network properties to evaluate the effectiveness of our proposed approach. The empirical results show that our approach can scale up to 10 times the problem size of Full-REMEDY in terms of planning time horizon. Meanwhile, it outperforms Fast-REMEDY by up to 33% in terms of the total number of healthy people over time. The robustness analysis shows that it works better than baselines even with network structure uncertainty. Interestingly, the policy analysis results show that compared with the baselines, our approach does not rely on node structural importance (e.g., degree and betweenness), and thus is fairer in the sense that it tends to spread the screening across different nodes.

1.2.2 Mobile Health Intervention

Mobile interventions are a model for providing services in which agents are sent to different locations where they provide various forms of interventions locally. Of particular importance are mobile health clinics (MHCs), a model of healthcare delivery in which mobile units deliver health services directly to target communities. MHCs are successful in reaching vulnerable populations; they overcome typical barriers to health services access, such as limited transportation, finances, insurance, or legal status Stephanie et al. [2017]. A wide variety of MHC services—such as primary care, prevention screenings, disease management, and treatment support—have been very successful. Their success is based on their flexibility in meeting the changing needs of target communities, and providing these services at discounted rates or free of charge. Compared to other healthcare service models, MHCs have been observed to provide cost savings and cost-effectiveness Stephanie et al. [2017]. Another important application of mobile interventions is in food pantry services, which cater to communities experiencing food insecurity by dispatching food trucks. We focus specifically on interventions for managing non-communicable diseases such as diabetes, cardiovascular disease, cancer, chronic respiratory disease, and mental health problems.

Network Restless multi-armed bandits Problems: These mobile intervention applications can be modeled as Restless multi-armed bandits (RMABs). RMABs have become a widely adopted mathematical model for studying various types of intervention services Kumar & Saranga [2010], Deo et al.

Figure 1.3: Example of mobile health service.

[2013], Mansour et al. [2015], Lee et al. [2019], Mate et al. [2020], Biswas et al. [2021], Xu et al. [2021]. RMABs are a model for sequential planning problems: in each round, a planner has to select k out of m arms to pull. Arms transition randomly between states, but the transition probabilities differ based on whether an arm was pulled or not. The arms dispense rewards depending on their state. In above applications, arms represent locations, k may represent the budget (e.g., number of available MHC units), and rewards are the number of people positively affected by an intervention. I extend existing RMAB models for interventions by considering network effects. Such network effects often arise due to individual commuting behavior: when an MHC visits one location, it provides interventions not only to people who reside there, but also to others who have traveled to this location (e.g., as a part of their routine work-related commuting). On the flip side, the same MHC may *miss* people who have traveled to a different location. Visiting one location may thus deliver an intervention to residents of multiple locations, giving rise to network effects.

Mobile Health Intervention-Contributions

Network effects lead to significant new challenges in the formal model. Common solution approaches for RMABs treat each arm as a Markov Decision Process (MDP) and exploit the fact that these MDPs are coupled only through the joint budget constraint. This weak coupling forms the basis for solutions based on index values, which are computed separately for each of the m arms. Policies that select the k arms with the highest indices can be shown to be asymptotically optimal for several domains Honda & Takemura [2010], Maillard et al. [2011], Kaufmann et al. [2012]. This thesis shows that the aforementioned network effects induce a stronger coupling between arms, making these solution approaches significantly less effective. The main contributions of this thesis toward MHC scheduling problem are (1) we present a class of RMAB models with network effects suitable for modeling mobile intervention domains, (2) we present a solution approach for this class of problems and provide sufficient conditions for the optimality of our approach, and (3) we show empirically that our solution delivers superior performance compared to existing approaches across multiple domains.

1.3 Thesis Outline

In Chapter 2, I discussed the related work for the network epidemic models, previous common practices of the health applications in my thesis, different sequential planning models, and some related network problems. Next, In Chapter 3, I introduce the active screening model I proposed and its solution for short term planning. Chapter 4 further extended the model for a long term planning solution by applying RL and overcome several challenges that traditional RL cannot handle. In Chapter 5, I presented the network mobile health intervention problem modeled as RMAB as another example of sequential network planning problem and its solution. Finally, In Chapter 6 I discuss the relevant future work for solving sequential planning problems on active screening, mobile health interventions and other public health applications and the challenges of applying them to the real-world and conclude my thesis.

2

Background and Related Work

2.1 Sequential Network Planning Problems for Public Health

Sequential resource allocation problems on networks constitute another active area of research in their applications in the real-world domain. The network effect can be roughly categorized into two types: (1) effect on state transitions (2) interventions that have network effects. Each type of network effect has its application in the public health domain. Due to the interaction between individuals or communications between locations in the real world, it is vital to harness the network structure as a whole instead of just single entries of data points.

My thesis provides an iconic application for each type of network effect. The first application is active screening, where the network effect arises from the physical contact between individuals. The main challenge of this problem comes from the network effect and the uncertainty of its state transition. The second application is mobile health intervention. In this application, communications between each location have caused a strong coupling effect of our intervention. Such coupling cannot be handled by previous standard approaches such as index policies due to the need to consider the set to intervene together instead of nodes to intervene individually. Chapter 4 of this thesis will illustrate more details, including examples.

2.2 Network Models in Public Health domain

In this section we discuss the network models used in previous literature of public health domain. One of the most popular model is the network epidemic Model. Epidemic models continue to be widely used across biological, social, and computer sciences. Applications range widely, including influence propagation Kempe et al. [2003], Yadav et al. [2016a], Wilder et al. [2017], rumor adoption Weenig & Midden [1991], computer virus suppression Garetto et al. [2003], and of course, disease spread. The studies of disease spreading history can date back to as early as 1760 when Bernoulli proposed the first mathematical epidemic model for smallpox (Variola Major) Bernoulli & Blower [2004]. In early 2000, studies Wang et al. [2003] have found that graph-based epidemic propagation models provide a more realistic approach compared to fully mixed models of earlier literature. Under these graph-based models, non-recurrent and recurrent disease suppression and eradication have been studied using different approaches.

2.2.1 Non-Recurrent Diseases

A large portion of work related to active screening deals primarily with SIR or SEIR type diseases (with two extra state *Exposed* and *Recovered*), often referred to as the *Vaccination Problem* Ball et al. [2015], Sun & Hsieh [2010], Wang [2005], Zhang & Prakash [2015], Ganesh et al. [2005], where permanent immunization (entry into R state) can be viewed as removing nodes from the graph. Exploiting this idea, Saha et al. [2015] and Tong et al. [2012] focus on immunization ahead of an epidemic and suggest a heuristic method of removing a set of k nodes based on the eigenvalues of the adjacency matrix. Zhang & Prakash [2015] consider the problem of selecting the best k nodes to immunize in a network after the disease has started to spread. Ren et al. [2018] extend the problem to tackle network with graph structure uncertainty. These methods do not apply to our scenario as they assume that a single round of screening offers permanent immunity.

2.2.2 Recurrent Diseases

For diseases in which there is no permanent immunity, one-time screening (cure) is not enough and, further, it may not be reasonable to quarantine patients until the disease has died out. When the true state of the graph in every round is known (in other words, when the policymaker has *perfect observations*), given certain budget constraints, Drakopoulos et al. [2016, 2014] provide a theoretical lower bound on the expected time needed to eradicate the disease, which grows linearly in the number of nodes. The authors provide a policy to show that disease eradication is possible when the graph structure and budget have specific properties under such perfect observation. Scaman et al. [2016] provide a scalable algorithm *maxcut minimization* and tighter theoretical bound of the eradication time based on the idea.

2.2.3 Comparison

My work differs from studies of recurrent diseases that assume perfect observations and seek to bound eradication time Drakopoulos et al. [2016, 2014], Scaman et al. [2016]. The impact of curing uncertainty in these previous works is analyzed in Hoffmann & Caramanis [2018] by providing non-constructive, algorithm-independent bounds, motivating this work. Chapter 2 of this thesis focus on developing algorithms to minimize the disease spread. This complex setting has not been studied previously. Although both inherently a multiagent problem because nodes (agents) make decisions in response to those around them, this problem of minimizing disease spread is different from another well-studied multiagent problem of influence maximization in general Kempe et al. [2003], Chen et al. [2009], Maghami & Sukthankar [2012], Yadav et al. [2016b], Wilder et al. [2018]. The influence maximization problem optimizes the selection of seeds or starting nodes for maximizing influence spread that usually has sub-modular property to exploit, as opposed to optimizing the selection of nodes on which to intervene to minimize disease spread.

2.3 Approaches for Solving Sequential Planning Problems

In this section we discussed the approaches for solving sequential planning problems in previous literature. We also discuss some of the new challenges emerged when applying these technique to the health applications with network effects we focused on.

2.3.1 Reinforcement Learning

Reinforcement Learning has attracted a lot of interest from researchers in the machine learning and artificial intelligence communities Mnih et al. [2015], Silver et al. [2016, 2017]. It is an experiment-driven and mathematical framework that trains an agent through trial and error Kaelbling et al. [1996], Sutton et al. [1998], Sutton & Barto [2018]. With the rise of deep learning, researchers further overcome the computational limitations of traditional RL by utilizing the representation power of deep neural networks Mnih et al. [2013], Arulkumaran et al. [2017], such as recurrent neural networks (RNNs) Zaremba et al. [2014], convolutional neural networks (CNNs) Krizhevsky et al. [2012] and graph convolutional neural networks (GCNs) Kipf & Welling [2017].

In Chapter 2 I address the challenges of uncertain states and limited budget by proposing the REMEDY algorithm. However, the variants of this approach either do not scale with the planning time horizon or fail to fully account for future actions. Due to the superior performance of RL approaches in solving long term planning problems Mnih et al. [2015], Silver et al. [2016, 2017], in Chapter 3 I propose a novel RL approach that builds upon a powerful variant of RL called DQN Mnih et al. [2013]. I first formulate the multi-round active screening problem as a Markov Decision Process (MDP), where the state is a vector representing the probability of each node in the network being infected, and the action is to select which subset of nodes to actively screen. Due to the extremely high-dimensional state and action spaces, vanilla DQN algorithms cannot be directly applied to solve our problem efficiently. I therefore design several innovative adaptations over vanilla DQN that fully exploit the problem structure of multi-round active screening. First, I show that the node features in the underlying contact network are inter-correlated. To efficiently capture the intrinsic correlations between different nodes, I use GCNs as the function approximator to represent the Q-function. Second, because in each time period we need to select a subset of nodes to actively screen, this leaves vanilla DQN un-scalable as it needs to solve a combinatorial optimization problem in the action selection procedure. To avoid this we decompose the node set selection problem in each time period as a sub-sequence of decisions, and then design a novel two-level RL framework that solves the problem in a hierarchical manner. It has two types of agents. The primary agent works at the main sequence level and interacts with the environment, while multiple secondary agents work at the sub-sequence level and are responsible for generating actions sequentially within each time period. Last, we find that the reward signals for the secondary agents are sparse. To speed up the slow convergence of secondary agents' policies that arises from the sparseness of rewards, we incorporate ideas from curriculum learning into our algorithm. Intuitively, the algorithm warm-starts at the beginning of training with a simpler task, which has limited action choice and true state information. As the training goes on, the algorithm gradually increases task difficulty by providing uncertain state information and more action choice until the problem becomes the same as the original active screening problem.

2.3.2 Restless Multi-Arm Bandit

Restless multi-armed bandits (RMABs) have become a widely adopted mathematical model for studying various types of intervention services Kumar & Saranga [2010], Deo et al. [2013], Mansour et al. [2015], Lee et al. [2019], Mate et al. [2020], Biswas et al. [2021], Xu et al. [2021]. RMABs are a model for sequential planning problems: in each round, a planner has to select k out of m arms to pull. Arms transition randomly between states, but the transition probabilities differ based on whether an arm was pulled or not. The arms dispense rewards depending on their state. In the motivating applications of Chapter 4, arms represent locations, k may represent the budget (e.g., number of available MHC units), and rewards are the number of people positively affected by an intervention. In Chapter 4 of this thesis, I extend existing RMAB models for interventions by considering network effects. Such network effects often arise due to individual commuting behavior: when an MHC visits one location, it provides interventions not only to people who reside there, but also to others who have traveled to this location (e.g., as a part of their routine work-related commuting). On the flip side, the same MHC may *miss* people who have traveled to a different location. Visiting one location may thus deliver an intervention to residents of multiple locations, giving rise to network effects.

Network effects lead to significant new challenges in the formal model. Common solution approaches for RMABs treat each arm as a Markov Decision Process (MDP) and exploit the fact that these MDPs are coupled only through the joint budget constraint. This weak coupling forms the basis for solutions based on index values, which are computed separately for each of the m arms. Policies that select the k arms with the highest indices can be shown to be asymptotically optimal for several domains Honda & Takemura [2010], Maillard et al. [2011], Kaufmann et al. [2012]. In Chapter 4, I show that the aforementioned network effects induce a stronger coupling between arms, making these solution approaches significantly less effective.

In the most general setting, the RMAB problem is known to be PSPACE-hard to solve optimally Papadimitriou & Tsitsiklis [1994]. However, by exploiting the problem structure of certain restricted classes of RMABs, efficient algorithms have been derived, sometimes with performance guarantees. The most popular of these is the Whittle index policy Whittle [1988] which is asymptotically optimal for *indexable* bandits Weber & Weiss [1990] and fast to compute if a closed form can be derived for the index. Many works are dedicated to proving the indexability of different RMAB subclasses and deriving closed-form or efficient approximations of the Whittle index Glazebrook et al. [2006], Mate et al. [2020], Hsu [2018], Akbarzadeh & Mahajan [2019]. Others have provided sufficient conditions for indexability Nino-Mora [2001] or developed expensive methods for computing policies with tighter reward bounds Bertsimas & Niño-Mora [2000], Adelman & Mersereau [2008]. However, all of these methods rely on the idea that the only factor coupling the arms are one or more budget constraints which we refer to as the *weakly coupled property*. Thus, previous RMAB methods will not be applicable for our work as the network effect strongly couples the states, actions, transitions, and rewards of neighboring arms.

In terms of applications, RMAB models have been widely used for scheduling problems, such as machine maintenance and repair Wang [2002], Abbou & Makis [2019], Glazebrook et al. [2006]. In these works, machines in factories are modeled as arms, and the goal is to find the optimal schedule to visit factories to maintain the machines. Other examples include anti-poaching patrol planning (Qian et al. [2016] propose a RMAB framework in which arms are poaching targets, and playing an arm corresponds to a patrol) or recommendation systems (e.g., for music streaming Zeng et al. [2016], Yi et al. [2017]). Such problems also motivated the recharging bandit model Kleinberg & Immorlica

[2018].

In this model, each arm's reward is determined by a function of the time elapsed since the arm was last pulled. Implicitly, this resets the arm's reward to time 0 whenever the arm is pulled. When these functions are increasing and concave for each arm, Kleinberg & Immorlica [2018] develop a concave program to solve the optimal frequency of pulling each arm; the program's value upper-bounds the value of an optimal schedule. Scheduling the arm then becomes a pinwheel scheduling problem Holte et al. [1989], and Kleinberg & Immorlica [2018] use a rounding scheme to approximate the scheduling of arm pulls, while obeying the frequency restriction. I extend this setting by allowing the arms' rewards to be only *partially* reset when the arm is selected, as well as by considering network effects.

In the public health domain, this work's focus, Mate et al. [2020] proposed collapsing bandits to improve medication adherence through interventions on patients. Lee et al. [2019] and Ayer et al. [2019] proposed RMABs for scheduling cancer screenings and hepatitis treatments, respectively. In Deo et al. [2013], the closest RMAB application to ours, the authors model the resource allocation problem of delivering school-based asthma care for children. The most important difference between my work and theirs is that my work consider network effects in the RMAB model.

3

Active Screening Problem: Short-term Planning

In this chapter, we discuss the REMEDY algorithm I developed to solve the active screening problem with short-term planning. The active screening problem is a sequential network planning problem with the network effect of the state transition. We will first discuss the detail of our network model in the following section.

	Notations for Model
Notation	**Definition**
S	susceptible state
I	infected state
β	transmission rate
γ	cure rate
t	time step number
T	terminal time step
k	budget for each time step
$\partial(v)$	set of v's neighbors
$s_v(t)$	state of v at time t
$C_a(t)$	set of nodes actively screened
$C_n(t)$	set of nodes naturally cured
$\mathbf{t_v}(t)$	true state vector of node v at time t
$\mathbf{T}_v^N(t)$	true state transition matrix for $V \setminus C_a(t)$
$\mathbf{T}_v^A(t)$	true state transition matrix for $C_a(t)$
	Notations for Algorithm
Notation	**Definition**
$\mathbf{b_v}(t)$	marginal belief state vector of node v at time t
$\bar{\mathbf{b}}_{\mathbf{v}}(t)$	intermediate belief state after knowing $C_n(t)$
$\mathbf{B}_v^N(t)$	transition matrix for $V \setminus C_a(t) \cup C_n(t)$
$\mathbf{B}_v^A(t)$	transition matrix for $C_a(t) \cup C_n(t)$
$x_v(t)$	marginal probability of v being in I state
$\mathbf{x}(t)$	probability vector of all nodes being in I state
$\mathbf{A}$	adjacency matrix of the graph
$\mathbf{R_a}(t)$	action choice matrix decided by the algorithm
$\mathbf{M_a}(t)$	upper bound transition matrix, function of $\mathbf{R_a}$
F	upper bound of objective function with true probability

Table 3.1: Notations Summery for this Chapter

3.1 Disease Model

We introduce the disease model for our problem, which is based on the well-known SIS model Anderson & May [1992], Bailey [1975]. An individual can either be in state S (a healthy individual *susceptible* to disease) or I (the individual is *infected*). SIS models capture the dynamics of recurrent diseases, where permanent immunity is not possible (e.g., TB, typhoid).

We adopt a discrete time SIS model for modeling the disease dynamics propagating on a graph. Given a contact network $G(V, E)$, infection spreads via the edges in the network. There are $|V|$ individuals, and we use $\partial(v)$ to denote neighbors of node v in the network. Each individual (node) v in the network at time t is in state $\mathbf{s}_v(t) \in \{S, I\}$. Let $\mathbf{t}_v(t)$ denote the state vector that represents the true state of node v at time t where S is represented as $[1, 0]^\top$ and I as $[0, 1]^\top$. Given the initial state, an infected node infects its healthy neighbors with rate β independently and recovers with probability γ. The latter term represents the probability that the node may visit a doctor on its own initiative. The health state transition probabilities of a node is then given by $P\left[s_v(t+1) = \{S, I\}\right] = \mathbf{T}_v^N(t)\mathbf{t}_v(t)$ where

$$\mathbf{T}_v^N(t) = \begin{array}{c c} & \begin{array}{cc} S & I \end{array} \\ \begin{array}{c} S \\ I \end{array} & \begin{bmatrix} 1 - q_v & \gamma \\ q_v & 1 - \gamma \end{bmatrix} \end{array}, \tag{3.1}$$

and $q_v = 1 - (1 - \beta)^{|\{u \in \partial(v) \mid \mathbf{s}_\mathbf{u}(t) = I\}|}$. The columns denote the state of v at time t and the rows denote the state at $t + 1$. The transition probabilities follow the disease dynamics described earlier. In particular, q_v captures the exact probability that node v becomes infected from its neighbors $\{u \in \partial(v) \mid \mathbf{s}_\mathbf{u}(t) = I\}$ and γ captures the probability that I individuals seek treatment voluntarily.

Given such transition probabilities and an initial state, if no intervention happens, the network state evolves by flipping biased coins for each node to determine their next true state in each round. The process is repeated until the terminal step T is reached.

3.2 The Active Screening (ACTS) Problem

Motivated by active screening/contact tracing campaigns that have been practiced since the 1980s Cadman et al. [1984] and applied in various forms/diseases Braxton et al. [2017], we propose the Active Screening (ACTS) Problem. Given the SIS model in the previous section, an active screening agent seeks to determine the best node sets $C_a(t) \subset V$ to actively screen and cure with a limited budget of $|C_a(t)| \leq k$ at each round t. The agent does not know the ground truth health state of all individuals. The agent knows the network structure $G(V, E)$, the infection probability β, and recovery probability γ. In addition, the agent observes the *naturally cured* node set $C_n(t)$ at time t—because this set of patients come to the clinic voluntarily. Active screening starts after the agent acquires information about $C_n(t)$. Let $C_a(t)$ be the set of nodes that are actively screened at time t. A node $v \in C_a(t)$ becomes cured at time $t+1$. Thus, the transition matrix for a node $v \in C_a(t)$ is $P[s_v(t+1) = \{S, I\}] = \mathbf{T}_v^A(t)\mathbf{t_v}(t)$, where

$$\mathbf{T}_v^A(t) = \begin{array}{c c} & \begin{array}{cc} S & I \end{array} \\ \begin{array}{c} S \\ I \end{array} & \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix} \end{array}. \tag{3.2}$$

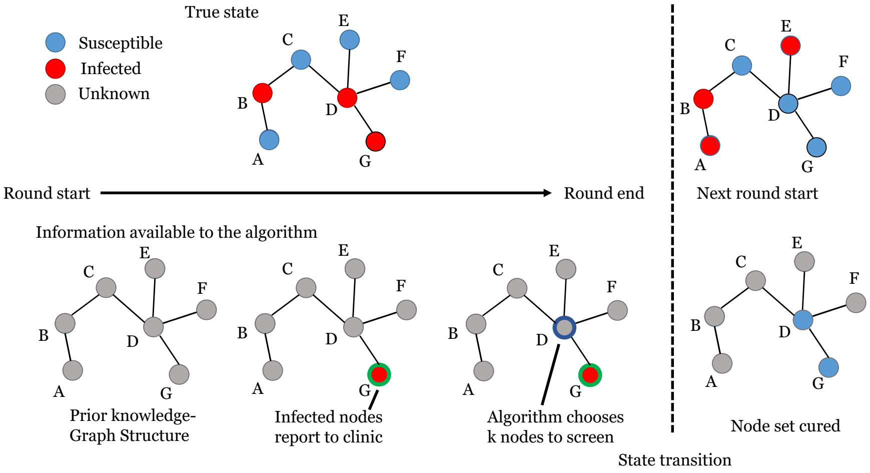


Figure 3.1: The procedure of the ACTS problem.

The action the agent takes at time t does not affect the transition matrix $\mathbf{T}_v^N(t)$ of the nodes not involved in active screening. Fig. 3.1 illustrates an example of the problem procedure. The upper part of the figure shows how the true state of the network evolves and the lower part of the figure shows the information available to the algorithm. In this example, there are seven nodes A$\sim$G. In each round, infected nodes (nodes B, D, and G in the example) flip a coin and report to the clinic with probability γ. The algorithm acquires the information of the nodes that eventually report to the clinic and are about to be cured, which is $\{G\}$ this round. Based on this information, the algorithm will choose a set of nodes, say $\{D\}$, to actively screen. These two sets of nodes are guaranteed to be in S state in the next round. After that, the state of the network transitions and the next round starts.

It is worth noting that although both the nodes that voluntarily report to the clinic and the nodes that are actively screened are guaranteed to be in S state in the next round, their neighbors may still be infected by them in the current round. In the example, node E is infected by node D even though node D was actively screened. This allows us to simplify the state transitions because curing and spreading infection occur at the same time.

Our objective is to maximize the health quality of each individual at each round (in contrast to past work, which primarily focuses on the cost of eradicating the disease entirely). The objective of the ACTS problem is:

$$\min_{C_a(0),\ldots,C_a(T)} \mathbb{E}\left[\sum\nolimits_{t=0}^{T} \sum\nolimits_{v\in V} 1_{\mathbf{s}_v(t)=I}\right]. \tag{3.3}$$

Problem Statement. *(ACTS Problem) Given a contact network $G(V,E)$, the disease and active screening model, find an active screening policy such that the expectation of $\sum_{t=0}^{T} \sum_{v\in V} 1_{\mathbf{s}_v(t)=I}$ is minimized.*

Even assuming we know the ground truth infected state for each node, ACTS is NP-hard. All proofs are in the supplemental material.

Theorem 1. *The ACTS Problem is NP-hard.*

Proof. We reduce the VertexCover decision problem "Is there a vertex cover of size k" to "Does there exist a curing strategy of objective function smaller or equal to $5|V| - 2k$ with budget of k each round of the constructed ACTS problem?"

Given a VertexCover decision problem with graph $G = (V, E)$ and budget k, we construct a new graph $G^* = (V_0^* \cup V_1^* \cup V_2^*, E^*)$ as follows: First, for each node $v \in V$, create three nodes v_0, v_1 and v_2 in G^*. Second, for each node $v \in V$, create an edge (v_0, v_1) in G^*. Finally, for each edge $(u, v) \in E$ create two edges, (u_1, v_2) and (u_2, v_1) in G^*. We set the parameters of the ACTS problem to be $(\beta, c) = (1, 0)$ and $T = 2$ with budget of k in each round. The initial state of the graphs are $s_v(0) = I \, \forall \, v \in V_0^*$ and $s_v(0) = S \, \forall \, v \in V_1^* \cup V_2^*$. Figure.3.2 shows a simple example.

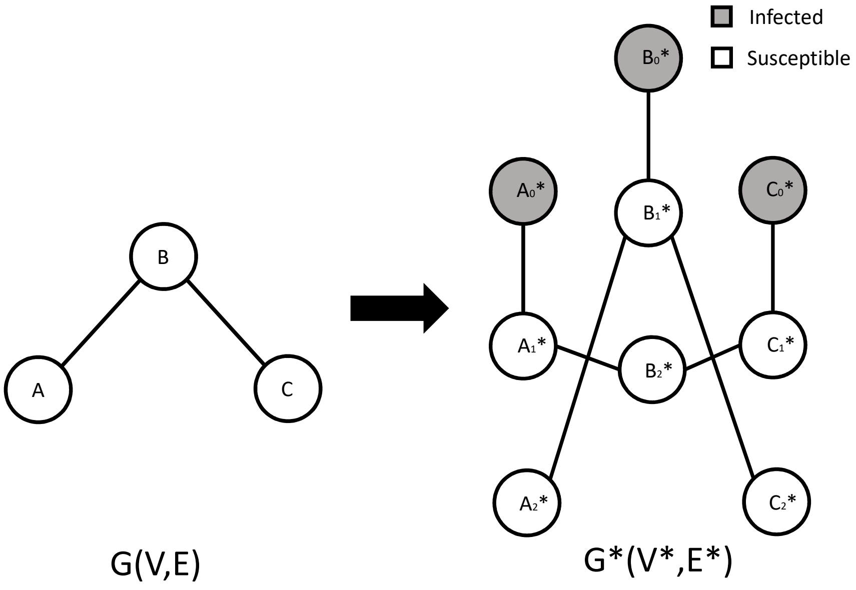


Figure 3.2: A simple example of graph transformation for problem deduction.

We now argue that G has a vertex cover of size k if and only if the ACTS problem of the above setting has the objective function smaller or equal to $5|V| - 2k$. In the above setting, we get to act twice. Acting at $t = 0$ allows us to force k nodes into S state at $t = 1$. Denote the objective function we get at time t as $Score(t)$, no matter what nodes we chose at $t = 0$, our sum of score in the first two rounds is always going to be $Score(0) = |V|$, $Score(1) = 2|V| - k$ and for the action we take at $t = 1$ will only reduce $Score(2)$ by amount of k, as long as we pick nodes in I state since it has no chance to propagate. Thus the only action matters is the action on $t = 0$ toward $Score(2)$. In the case where k equals to minimum vertex cover, picking the copy of vertex cover set of G in V^*_1 results in $|V| + (|V| - k) + k$ of I nodes in $t = 2$, which are all the nodes in V^*_0, all the nodes in V^*_1 except vertex cover copy and the vertex cover in V^*_2. We argue that this is the optimal strategy as picking anything that is not vertex cover results more than k infected nodes in V^*_2. Then we pick arbitrary k nodes as our action in $t = 1$ and results a score of $Score(2) = 2|V| - k$. For the case where k is larger then minimum vertex cover, containing any vertex cover will result in an objective function smaller then the $5|V| - 2k$ threshold. The intuition is the infected node in V^*_2 set is either result in (1) some of its edges are not covered (2) the node itself is covering the edge. In the case that an arbitrary vertex cover is picked, (1) will not happen and (2) is always smaller (if some picked nodes are fully covered by other nodes already) or equal (otherwise) to k. Thus one can always achieve an objective function less then the threshold. For budget size smaller then minimum cover, it is clear that the infection of layer V^*_2 is always going to result in an objective function higher then threshold since there must be some edges are not covered. Thus checking this threshold determines if the vertex cover exist or not. Thus we have proven the ACTS problem to be NP-hard. □

We introduce REMEDY, a software agent for assisting to select nodes to actively screen in the ACTS problem. REMEDY, shown in Algorithm 1 has two components: (i) a marginal belief state update that we use for reasoning about the infected status of nodes, and (ii) an algorithm for selecting which nodes to actively screen based on the marginal belief state and an upper bound of the ACTS objective.

Figure 3.3 shows how REMEDY observes and interacts with environment repeatedly. In each

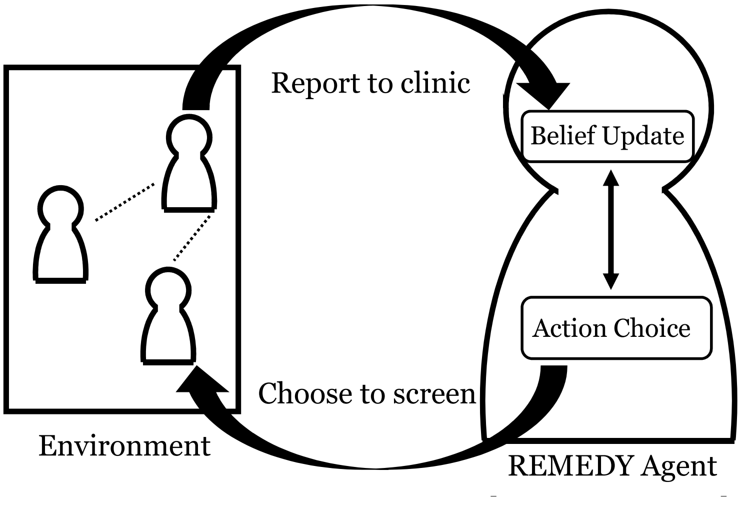


Figure 3.3: REMEDY overview

round, agents who are naturally cured $(C_n(t))$ in the environment (left rectangle) report to the clinic and are observed by REMEDY. After REMEDY perceives such information, it uses (i) to update its belief. Then, based on the current belief, it determines a set of agents to actively screen $(C_a(t))$ as its action by (ii). Finally, based on the action it takes and its prediction of environment transition, it updates its belief again by (i) and is ready for the next round observation.

3.2.1 Belief State Update

Tracking the exact probability that a node is infected in ACTS requires storing $O(2^{|V|})$ values, which is computationally intractable for reasonably sized graphs. Thus, REMEDY maintains a belief state based on the *marginal* probability that each node is infected, requiring only $O(|V|)$ values for storage. To calculate the marginal infection probability for the next round, we have to consider all possible events of a node's neighbors are infected or not, which appears to require computing a sum with exponentially many terms. We prove in Lemma 1 that the sum may be written with a linear number of terms. However, the marginal belief state discards correlation between nodes and this may lead to underestimating the number of infected nodes. We address this issue in the next section by deriving an upper bound for the true ACTS in terms of the marginal belief state. we form an upper bound on the ACTS objective that accounts for the imprecision of the marginal belief state. Fig. 3.3 illustrates the procedure of our algorithm.

The marginal belief update is lines 1–7 and 9–15 of Alg. 1. At each round $t \in \{0, \ldots, T-1\}$, we acquire perfect information about the infected state of each I node that naturally recovers, i.e., the nodes that satisfy $s(t) = I$ and $s(t+1) = S$. The state of the remaining nodes is unknown.

Let $x_v(t) \in [0,1]$ be the probability that node v is in state I at time t, and let $\mathbf{b_v}(t) = [1 - x_v(t), x_v(t)]^\top$ be the marginal belief vector. For each node, we update an intermediate belief state $\overline{\mathbf{b}}_\mathbf{v}(t) = [1 - \bar{x}_v(t), \bar{x}_v(t)]^\top$ in which $\bar{x}_v(t) = 1$ for $v \in C_n(t)$ and $\bar{x}_v(t) = \frac{(1-\gamma)x_v(t)}{(1-x_v(t))+(1-\gamma)x_v(t)}$ for the remaining nodes $v \in V \setminus C_n(t)$. These update steps are in lines 1–7 of Algorithm 1. This intermediate belief state is then exploited by the action choice subroutine to select $C_a(t)$, the node set we actively cure (line 8). After that, we calculate the marginal belief state at the next round: $\mathbf{b_v}(t+1) = \mathbf{B}_v^N(t)\overline{\mathbf{b}}_\mathbf{v}(t)$ and $\mathbf{b_v}(t+1) = \mathbf{B}_v^A(t)\overline{\mathbf{b}}_\mathbf{v}(t)$ for $v \in V \backslash (C_n(t) \cup C_a(t))$ and $v \in C_n(t) \cup C_a(t)$ respectively where

$$\mathbf{B}_v^N(t) = \begin{array}{cc} & \begin{array}{cc} S & \;\; I \end{array} \\ \begin{array}{c} S \\ I \end{array} & \begin{bmatrix} 1-p_v & 0 \\ p_v & 1 \end{bmatrix} \end{array}, \mathbf{B}_v^A(t) = \begin{array}{cc} & \begin{array}{cc} S & I \end{array} \\ \begin{array}{c} S \\ I \end{array} & \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix} \end{array} \tag{3.4}$$

and $p_v = 1 - \prod_{u \in \partial(v)}(1 - \beta \bar{x}_u(t))$. These steps are shown in lines 9-15 of Alg. 1. The transition matrix $\mathbf{B}^N$ does not contain parameter γ because each node in the I state that did not naturally recover will remain in I state with probability 1. It is worth noting that, intuitively, to update the marginal belief state for node v, one has to calculate the probability of all possible event of its neighbors being infected, which scales exponentially to the number of neighbors. Note that p_v is not an approximation but the exact value calculated by listing all possible events of v's neighbor being infected or not which we show in Lemma 1 that scales exponentially ($2^{|\partial(v)|}$). We show below in Lemma 1 that the approach adopted by Eq. 3.4 to calculate p_v yields the exact probability of v becoming infected by its neighbors given it is currently in S which saves a great amount of computational time.

Lemma 1. *The exact marginal probabilities of* $P[s(t+1) = I|s(t) = S]$ *can be calculated by* p_v *without listing the probability associated with each possible set of infected neighbors.*

Proof. The theorem can be proved by induction. For the base case where there is only one neighbor, the probability that node v is infected in the next time step given it is currently in S is $p_{v,1} = \bar{x}_u(1 - (1-\beta)^1) + (1-\bar{x}_u)(1-(1-\beta)^0) = \beta\bar{x}_u$. Assume $p_{v,k} = 1 - \prod\limits_{u\in\partial(v)} (1-\beta\bar{x}_u^t)$ for $|\partial(v) \leq k|$ is true, for $|\partial(v) = k+1|$, where w denotes the newly added neighbor, we have:

$$
\begin{aligned}
p_{v,k+1} &= p_{v,k} + \bar{x}_w\beta - p_{v,k}\bar{x}_w\beta \\
&= (1-\bar{x}_w\beta)(1 - \prod_{u\in\partial(v)\setminus w} (1-\beta\bar{x}_u)) + \bar{x}_w\beta \\
&= 1 - \prod_{u\in\partial(v)} (1-\beta\bar{x}_u)
\end{aligned}
$$

Thus we proved that p_v evaluates the exact probability of $P[s_v(t+1) = I|s_v(t) = S]$. □

3.2.2 Action Choice Algorithm

Possible approaches: We now turn our attention to selecting the set of nodes to actively screen, i.e., line 8 in Alg. 1. First, treating the Acts problem as a POMDP and applying state of the art reinforcement learning techniques is not feasible for the real world scenario we are aiming for Mnih et al. [2015].

Algorithm 1 REMEDY

```
Input: $\mathbf{A}, \mathbf{b}(t), \beta, \gamma, C_n(t), t, T, k$
Output: $C_a(t), \mathbf{b}(t+1)$

for $v \in V$ do
    if $v \in C_n(t)$ then
        $\bar{\mathbf{b}}_\mathbf{v}(t) \leftarrow [0, 1]^\top$
    else
        $\bar{\mathbf{b}}_\mathbf{v}(t) \leftarrow \frac{[(1-x_v(t)),(1-\gamma)x_v(t)]^\top}{((1-x_v(t))+(1-\gamma)x_v(t))}$
    end if
end for
$C_a(t) \leftarrow \textsc{ActionChoice}(\mathbf{A}, \bar{\mathbf{b}}(t), \beta, \gamma, C_n(t), t, T, k)$
for $v \in V$ do
    if $v \in V \setminus C_n(t) \cup C_a(t)$ then
        $\mathbf{b}_\mathbf{v}(t+1) \leftarrow \mathbf{B}_v^N(t)\bar{\mathbf{b}}_\mathbf{v}(t)$
    else
        $\mathbf{b}_\mathbf{v}(t+1) \leftarrow \mathbf{B}_v^A(t)\bar{\mathbf{b}}_\mathbf{v}(t)$
    end if
end for
return $C_a(t), \mathbf{b}(t+1)$
```

This is due to the fact that the computation time scales poorly with the high dimension action choice, which is exponential in the budget for our problem. Even when we approximated the actual feasible action choice by choosing the nodes greedily one node at a time and estimating the reward function, the resulting approach performed poorly and did not scale up to 20 nodes, which is less than even the smallest graph in our dataset.

One fast yet naive approach to this problem is to select the node set with maximum marginal belief to be in I state. This approach can be computed in $O(|V|)$, but it does not take the network structure and future infection probabilities into account. For example, suppose we have a tree structure with a known infection state: the root is the only infected node. The belief-based approach will screen the root and spend the remainder of the budget on random nodes. This is suboptimal because the remaining budget could be spent on the children of the root to prevent the disease from spreading.

Another approach is to choose nodes based on the graph structure. For example, we can select the nodes that, if deleted (permanently actively screened), would reduce the largest eigenvalue of the graph the most Prakash et al. [2012]. This approach guarantees that the infection is eradicated in the long term if the largest eigenvalue can be reduced below $\frac{\gamma}{\beta}$ for sufficient budget k. However, structure based approaches perform poorly when there are many nodes with identical roles in the graph structure, e.g., in symmetric graphs. Here, belief information would be more useful because it takes into account current signals from local neighbors of nodes

However, belief-based approaches and structure based approaches do not work well individually. A simple example where belief based approach fail would be a tree structure where its root is being infected, in which the belief based approach would not actively screen the children of infected node to prevent them being infect next round. As for structure based approaches, it performs poorly when the structure is rather symmetric like cube or circle graph. These method would just cure nodes randomly without considering belief information.

Although there are many heuristic ways one can combine belief and structure based approach, it is usually difficult to estimate the performance beforehand or derive the reasoning behind. These method may work well on some settings yet fail on another. We compare the state of the art CutWidth method for known state combined with belief state in the experiments.

Remark that though we store the marginal belief state, the upper bound is taken w.r.t. to the *true* ACTS objective.

Our approach: The key novelty of our software agent is that it brings together three key features: the use of belief states, a Frank-Wolfe style gradient-based algorithm for efficient reasoning about the structure of the graph, and use of an upper-bound of the *true* ACTS objective. Whereas algorithms for active screening have typically used discrete reasoning such as Markov chain (see Related work) and have not appealed to gradient-based approaches, it is the novel combination of this gradient-based approach with the use of belief states and upper bounds that is key in our work. Note that whereas marginal belief states avoid the exponential storage requirement of exact belief states, they typically underestimate the expected number of infected nodes as a result of the lost correlation information. We rely on using an upper bound on true number of infected nodes reduces this effect — thus we face the issue of determining a suitable upper bound. Our desiderata for determining this upper bound are therefore: (i) encapsulate the observations and actions of past and future, (ii) provide a performance guarantee compensating the information lost from marginal belief state, and (iii) be minimizable in time polynomial in T, k and $|V|$.

We develop two different algorithms for action choice: Full-ActionChoice, which looks ahead through all future actions and Fast-ActionChoice, a less computationally intensive variant that considers only the current action, allowing it to exploit eigenvalue decomposition. We refer to REMEDY agent using Full and Fast-ActionChoice as Full and Fast-REMEDY. Both Full-REMEDY and Fast-REMEDY, we change the action based on the observation in each round.

The key idea is to derive an alternative upper bound of the *true* ACTS objective. By establishing this function, we avoid the pitfall of directly optimizing the problem of NP-hardness and reduce the effect of often underestimated marginal belief state due to correlation information lost at the same time. Our desiderata for the upper bound function are (1) encapsulates the observations and actions of past and future (2) provides a theoretical guarantee of performance of any action choice (3) being mathematically sound for efficient computing that scales at most poly-nominal to t, k and $|V|$. It is not naive to derive such upper bound since we need to encapsulates the observations and actions of past and future while maintaining the function to be mathematically workable for efficient computing.

We start with some preliminary notation. To encapsulate the effect of active-screening toward our objective function, we define the $|V| \times |V|$ diagonal action matrix $\mathbf{R}_a(t)$ at time t as $\mathbf{R}_a(t)_{v,v} = 1$ if and only if $v \in C_a(t)$, and 0 otherwise. For the current round, say t_0, we observe the nodes that are cured and need to decide the nodes to actively screen. We define the *naturally cured matrix* $\mathbf{R}_n(t_0)$

as $\mathbf{R}_n(t_0)_{v,v} = 1$ if and only if $v \in C_n(t_0)$, which encapsulates the knowledge we gain from natural recovery in the current round. Let vector $\mathbf{x}(t)$ represent $x_v(t)$ for all v. To bound $\mathbf{x}(t)$ across all rounds given the actions we take, let $\mathbf{M}' = \beta\mathbf{A} + \mathbf{I}$, where $\mathbf{A}$ is the adjacency matrix and $\mathbf{I}$ is the identity matrix, define the *upper bound transition matrix* for the current round ($t = t_0$) as $\mathbf{M}_a(t_0) = (\mathbf{I} - \mathbf{R}_a(t_0) - \mathbf{R}_n(t_0))\mathbf{M}'$. And for future rounds ($t > t_0$), we define it as $\mathbf{M}_a(t) = (\mathbf{I} - \mathbf{R}_a(t))\mathbf{M}$ where $\mathbf{M} = \beta\mathbf{A} + (1-\gamma)\mathbf{I}$.

Theorem 2. *Let the current time be t_0. $\mathbf{M}_a$ is defined as above for t_0 and $t > t_0$. The ACTS objective (Eq. 3.3) is bounded above by:*

$$\mathbb{E}[\sum_{t=t_0}^{T}\sum_{v\in V}|s_v(t) = I|] \leq F = 1^\top \sum_{t=t_0}^{T}\prod_{\tau=t_0}^{t}\mathbf{M}_a(\tau)\mathbf{x}(t_0) \tag{3.5}$$

$$\textit{where} \prod_{\tau=t_0}^{t}\mathbf{M}_a(\tau) = \mathbf{M}_a(t)\mathbf{M}_a(t-1)...\mathbf{M}_a(t_0). \tag{3.6}$$

Proof. Given the marginal probability of node v and its neighbors, the exact conditional probability of $P\left[\mathbf{s_v}(t+1) = I|\mathbf{s_v}(t) = S\right]$ is bounded by:

$$P\left[\mathbf{s_v}(t+1) = I|\mathbf{s_v}(t) = S\right] \leq 1 - (1-\beta)^{\sum_{u\in\partial(v)} x_u}.$$

Since

$$\begin{aligned} P[\mathbf{s_v}(t+1) = S|\mathbf{s_v}(t) = I] &= 1 - P[\mathbf{s_v}(t+1) = S|\mathbf{s_v}(t) = S] \\ &= 1 - \sum_{m=0}^{|\partial(v)|} p_m (1-\beta)^m \\ &= 1 - \mathbb{E}[(1-\beta)^m] \\ &\leq 1 - (1-\beta)^{\mathbb{E}[m]} \\ &= 1 - (1-\beta)^{\sum_{u\in\partial(v)} x_u} \end{aligned}$$

We further approximate the right hand side by a first order Taylor series expansion as $\beta \sum_{u\in\partial(v)} x_u(t)$, yielding

$$\begin{aligned} x_v(t+1) = &(1 - x_v(t))P[\mathbf{s_v}(t+1) = I|\mathbf{s_v}(t) = S] \\ &+ x_v(t)P[\mathbf{s_v}(t+1) = I|\mathbf{s_v}(t) = I] \\ &= (1 - x_v(t))(1 - (1-\beta)^{\sum_{u\in\partial(v)} x_u}) + x_v(t)(1-\gamma) \\ &\leq (1 - x_v(t))\beta \sum_{u\in\partial(v)} x_u(t) + x_v(t)(1-\gamma). \end{aligned}$$

Using a vector $\mathbf{x}(t)$ to represent $x_v(t)$ for all v, the above yields the following equation in vector form:

$$\mathbf{x}(t+1) \leq \mathbf{M}\mathbf{x}(t) - diag(\beta\mathbf{A}\mathbf{x}(t))\mathbf{x}(t), \tag{3.7}$$

where $\mathbf{M} = \beta\mathbf{A} + (1-\gamma)\mathbf{I}$ and $\mathbf{A}$ is the adjacency matrix. We drop the negative term $diag(\beta\mathbf{A}\mathbf{x}(t))\mathbf{x}(t)$ and only consider $\mathbf{M}\mathbf{x}(t)$ as the upper-bound.

While the above holds without intervention, we need the form of matrix $\mathbf{M}$ with intervention and knowledge of $C_n(t_0)$. Suppose a node v is naturally cured or actively screened at time t ($v \in C_n(t) \cup$

$C_a(t)$), it is guaranteed to be in S state in $t+1$. Thus it does not accumulate any probability of being infected from its neighbor in time t nor does it delivers any probability of being infected toward its neighbors in time $t+1$. The former is equivalent to deleting the column v of $\mathbf{M}$ at time t and the latter is equivalent to deleting the row v of $\mathbf{M}$ at time $t+1$, which are $(\mathbf{I} - \mathbf{R}_a(t))\mathbf{M}$ and $\mathbf{M}(\mathbf{I} - \mathbf{R}_a(t))$ respectively for $t > t_0$. The two matrix could be combined while the matrix multiplication since $\mathbf{M}(\mathbf{I} - \mathbf{R}_a(t))^2\mathbf{M} = \mathbf{M}(\mathbf{I} - \mathbf{R}_a(t))\mathbf{M}$ as $(\mathbf{I} - \mathbf{R}_a(t))$ is a 0-1 diagonal matrix. Similar equation can be derived for $\mathbf{M}_a(t_0)$ and encapsulate the knowledge of $C_n(t)$ in $\mathbf{R}_c(t)$ as well. Thus we have $\mathbf{x}(t+1) \leq \mathbf{M}_a(t)\mathbf{x}(t)$ for all $t \geq t_0$ and finally $\sum_{t=t_0}^{T} 1^\top \mathbf{x}(t) \leq 1^\top \sum_{t=t_0}^{T} \prod_{\tau=t_0}^{t} \mathbf{M}_a(\tau)\mathbf{x}(t_0)$ yields to the result. □

Given that the function F upper bounds our objective function, we next describe the method we use to select the action matrix $R_a(t)$ that minimizes F for every round. Distinct from previous literature, our objective takes into account the number of infected nodes at each round. We also have the flexibility to change the action we take based on the observation we make in each round. Such flexibility results a solution space of size $\binom{|V|}{k}^T$, making the bound challenging to optimize exactly, since it is nonconvex. Hence, we apply a Frank-Wolfe style method Frank & Wolfe [1956] to the continuous relaxation. The result is Full-ActionChoice (Alg. 2), a gradient-based algorithm that runs for L iterations, simultaneously updating the actions taken at each round. It begins with an arbitrary feasible point and performs three steps per iteration: (i) computes the gradient of the objective at the current point, (ii) optimizes the linear approximation to the objective over the true (not relaxed) feasible set, and (iii) steps toward it. After L iterations, we greedily round the solution, selecting the k nodes that have highest values. Each time we receive a new naturally cured set, we run Alg. 2 over all remaining rounds and output the action for the current time.

Given the gradient and the naturally cured node set, an approximately optimal action for all times in the continuous relaxation can be obtained through a projected gradient descent or a Frank-Wolfe

Algorithm 2 Full-ActionChoice

```
**Input**: $\mathbf{A}, \overline{\mathbf{b}}(t_0), \beta, \gamma, T, t_0, k$
**Output**: $C_a(t_0)$

 1: $\mathbf{R}_a^0(t) \leftarrow 0 \quad \forall t$
 2: for $l = 1...L$ do
 3:     for $t = t_0...T$ do
 4:         $\Delta(t) \leftarrow$ GradientOracle$(\mathbf{R}_a^{l-1})$
 5:         $\mathbf{R}_a^*(t) \leftarrow$ ProjectFeasible$(\Delta, k)$
 6:         $\mathbf{R}_a^l(t) \leftarrow (1-\alpha_l)\mathbf{R}_a^{l-1}(t) + \alpha_l \mathbf{R}_a^*(t)$
 7:     end for
 8: end for
 9: $C_a(t_0) \leftarrow \arg\max_k \mathbf{R}_a^L(t_0)$
10: return $C_a(t_0)$
```

style algorithm, yielding Full-ActionChoice (Alg. 2).

We describe Alg. 2 in more detail. We initialize to an arbitrary feasible point in Υ, the convex hull of the binary valued $\mathbf{R}_a(t)$: we choose $\mathbf{R}_a^0(t) = 0$ for $t = t_0 \sim T$ in iteration $l = 0$ (line 1). We then update the candidate solution for every time step simultaneously in each iteration l in three steps. In each iteration, we need to caluclate the gradient of F w.r.t. the action choice, which is the GradientOracle of line 4. We relax the optimization to the continuous problem by allowing $\mathbf{R}_a(t)_{v,v}$ to take real values between 0 and 1, which can be interpreted as the probability of choosing node v. The feasible solution space is the convex hull of the binary valued $\mathbf{R}_a(t)$. We denote this convex hull Υ. By taking the derivative of F, the gradient w.r.t. action at each time t is

$$\frac{\partial F}{\partial \mathbf{R}_a(t)} = -\sum_{t'=t+1}^{T} \prod_{\tau=t'}^{t+1} \mathbf{M}_a^\top(\tau) \mathbf{1} \mathbf{x}^\top(t_0) \prod_{\tau=t-1}^{t_0} \mathbf{M}_a^\top(\tau), \tag{3.8}$$

The above gradient is a matrix $\Delta(t)$, where the diagonal elements $\Delta(t)_{v,v}$ represent the gradient w.r.t. the choice of node v to actively screen at time t.

We then minimize this linear approximation over the true feasible set. Since the objective is linear and the only constraints are individual variable bounds and the budget constraint, we can optimize exactly by greedily selecting the k nodes with largest $\Delta(t)_{v,v}$ as our current best solution $\mathbf{R}_a^*(t)$ in line 5. We set the initial point $\mathbf{R}_a^l(t)$ of the next iteration in line 6, in which $\alpha_l = 2/(l+2)$ is the step size of Frank-Wolfe algorithm. Since Υ is convex and $\mathbf{R}_a^l(t)$ is the convex combination of two feasible points, it is guaranteed that it will remain in the convex hull Υ after the update. After L iterations, we output our action in the current round by greedily selecting k nodes of the relaxed $\mathbf{R}_a^L(t_0)$ of the final iteration, as line 9 shows.

The Full-REMEDY algorithm considers future actions simultaneously and has time complexity of $O(T^2|V|^\omega)$, where the exponent ω arises from complexity of matrix multiplication (best known ω is around 2.37). The algorithm used scales well to the budget k. However, calculating such solutions for a very large network—which is often the case for active screening—can be time consuming. To reduce time complexity, we further simplify the upper-bound function by assuming that no actions are taken in the future rounds and ignore their effect on the current decision making in Fast-ActionChoice (Alg. 3). By ignoring future actions, the action matrix $M_a(t)$ in Full-REMEDY is simplified to constant M. The contribution of actively screening each node can be written as the following vector form:

$$1^\top \sum\nolimits_{\tau=0}^{T-t_0-1} \mathbf{M}^\tau diag(\mathbf{M}_n \mathbf{x}(t_0)), \tag{3.9}$$

where $\mathbf{M}_n = (\mathbf{I} - \mathbf{R}_n(t_0))\mathbf{M}'$. Now, since $\mathbf{M}$ is the same for every future round, $\mathbf{M}$ can be decomposed as $\mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^\top$ ahead of time, where $\mathbf{Q}$ is a matrix comprised of the eigenvectors of $\mathbf{M}$, and $\boldsymbol{\Lambda}$ a diagonal matrix comprised of the eigenvalues along the diagonal. Such a matrix can be approximated by calculating only the top m largest eigenvalues and their eigenvectors using the Lanczos algorithm Lanczos [1950] that has a complexity of $O(|E|)$ (assuming the large network is sparse), yielding the

Algorithm 3 Fast-ActionChoice

```
Input: $\mathbf{A}, \overline{\mathbf{b}}(t_0), \beta, \gamma, T, t_0, k$
Output: $C_a(t_0)$

1: if $t_0 = 0$ then
2:     $\mathbf{M} \leftarrow \beta\mathbf{A} + (1-\gamma)\mathbf{I}$
3:     $\mathbf{Q}_m, \mathbf{\Lambda}_m \leftarrow$ Lanczos$(\mathbf{M}, m)$
4: end if
5: Scores $\leftarrow 1^\top \mathbf{Q}_m(\sum_{\tau=0}^{T-t_0-1} \mathbf{\Lambda}_m^\tau)\mathbf{Q}_m^\top diag(\mathbf{M}_n\mathbf{x}(t_0))$
6: $C_a(t) \leftarrow k$ nodes with highest scores in vector Scores
```

Fast-ActionChoice shown in Alg. 3. The approximate $\mathbf{M}$ is given by $\mathbf{Q}_m\mathbf{\Lambda}_m\mathbf{Q}_m^\top$, where these matrices are computed in line 3. In line 5, the well-known result $(\mathbf{Q}_m\mathbf{\Lambda}_m\mathbf{Q}_m^\top)^\tau = \mathbf{Q}_m\mathbf{\Lambda}_m^\tau\mathbf{Q}_m^\top$ is used to approximate $\mathbf{M}^\tau$. The time complexity of Fast-REMEDY is $O(|V|^2)$ assuming constant m.

3.3 Experiments

We perform experiments comparing Fast- and Full-REMEDY to baselines on a variety of real-world datasets. Table 3.3 lists the networks and their properties. Most of the networks were collected in human contact settings. The networks are carefully selected to have significant variation in size($|V|$ ranging from 75 to 75879 nodes), average degrees (d), average shortest path length (ρ_L), assortativities (ρ_D) and epidemic thresholds ($1/\lambda_A$), which is also known as spectral radius.

Setting. Unless explicitly stated otherwise, we assume the budget k allows for screening and treatment of 20% of the total population $|V|$ per round. All results are averages over 30 runs.

In practice, active screening is performed only after conducting initial surveys on the prevalence and incidence of the disease. To simulate this, we run experiments in two stages.

Stage 1 (Survey Stage). This stage starts at $t = 0$ with 25% of individuals in I, selected uniformly at random, and ends at $t = 10$. No active screening is done and the disease evolves naturally. The

Table 3.2: Properties of the contact network data sets.

Network	$\|V\|$	$\frac{1}{\lambda_A^*}$	d	ρ_L	ρ_D
Hospital Vanhems et al. [2013]	75	0.027	15.19	1.60	-0.18
India Banerjee et al. [2013]	202	0.095	3.43	3.11	0.02
Face-to-face Isella et al. [2011]	410	0.042	6.74	3.63	0.23
Flu Salathé et al. [2010]	788	0.003	150.12	1.62	0.05
Irvine Panzarasa et al. [2009]	1893	0.021	7.29	3.06	-0.18
Escorts Rocha et al. [2010]	16730	0.032	2.33	4.20	-0.03

initial belief $\mathbf{b}(0)$ for all nodes is assumed to be $[0.5, 0.5]^\top$ since we have no prior information. Beliefs are updated according to the belief update algorithm in the *Disease model* section. This belief update requires knowledge of β and γ. There is a rich literature of how to estimate the disease parameters (β and γ) in this stage and these methods have been tested on real-world scenarios Kirkeby et al. [2017], Saad-Roy et al. [2016], Dong et al. [2012]. Here, we assume that such parameters are known.

Such parameters can vary from disease to disease. For example, the transmission rate of Pertussis can be as high as 0.47 for certain age groups Hethcote [1997], and as low as 0.035 for Syphilis Saad-Roy et al. [2016]. The cure rate also depends on how resourceful the target regions are. We initially assume $(\beta, \gamma) = (0.1, 0.1)$ and then evaluate a range of values.

Stage 2 (ACTS Stage). Here, we consider various screening algorithms. We perform active screening from $t = 11$ to $t = T = 20$ to represent 5 years of time (each round is 6 months CDC [2011]). Beliefs are updated according to the belief update scheme presented in *Disease Model and Background* section.

3.3.1 Metrics

We compare the outcomes of the following screening strategies compared to no intervention(**None**). In **None**, the evolution of the health states is based on disease dynamics only, with no active screen-

Table 3.3: Improvement over **None** in terms of the number of reduced infections (the larger the better). All computations are carried out with $\beta = 0.1, \gamma = 0.1$. Here, TLE signifies that the 24 hour limit was exceeded.

Network	Number of reduced infections						
	Random	Max-Degree	Eigenvalue	Max-Belief	BeliefCutWidth	Fast-REMEDY	Full-REMEDY
Hospital Vanhems et al. [2013]	144	150	151	150	147	**156**	**160**
India Banerjee et al. [2013]	605	470	420	636	754	**890**	**901**
Face-to-face Isella et al. [2011]	809	843	745	1057	1100	**1297**	**1409**
Flu Salathé et al. [2010]	1336	1421	1431	1438	1396	**1443**	**1446**
Irvine Panzarasa et al. [2009]	4630	5741	3692	4957	5623	**6676**	**7821**
Escorts Rocha et al. [2010]	27400	30167	TLE	29493	TLE	**46549**	TLE
Epinion Leskovec & Krevl [2014]	187369	228174	TLE	207565	TLE	**285280**	TLE

ing for all T rounds. The improvement over **None** is reported as the number of fewer infections as compared to **None**. Thus, the larger this number the better the performance of the algorithm.

(1a) Random: Randomly select nodes for active screening.

(1b) MaxDegree: Successively choose nodes with the largest degree until the budget is reached. This baseline uses only the graph structure information and thus does not update the belief state.

(1c) Eigenvalue: Greedily choose nodes that reduce the largest eigenvalue of **A** the most until the budget is reached.

(1d) MaxBelief: Choose nodes with the highest probability of being in the I state.

(1e) BeliefCutWidth: A modified version of the CutWidth method for a problem with known infection state Scaman et al. [2016], Drakopoulos et al. [2014]. Since the original method requires known infection state, we modified it by using a sample from the marginal belief state as a substitute of the true state. Note that due to the uncertainty of the network state, cure latency of active screening and budget limit, this baseline does not guarantee to eradicate the disease eventually.

Unfortunately, the data sets from countries that have high infectious disease burden, for which the algorithms may be applied on the ground in the future, have restrictive terms of use. For privacy and security reasons, they cannot be shared externally. Instead, we present the result of testing these algorithms on the following realistic contact networks collected from diverse sources.

(2a) **Hospital** Vanhems et al. [2013]: A dense contact network collected in a university hospital to study the path of disease spread.

(2b) **India** Banerjee et al. [2013]: A human contact network collected from a rural village in India where active screening with limited budget may take place.

(2c) **Face-to-face** Isella et al. [2011]: A network describing face-to-face contact in which influenza might spread through the close contact of individuals.

(2d) **Flu** Salathé et al. [2010]: A network of close proximity interactions in an American high school. The network is highly dense ($\lambda_A > 300$) with small-world properties and a relatively homogeneous degree distribution.

(2e) **Irvine** Panzarasa et al. [2009]: A friendship network collected from students in UC Irvine, used to study rumor modeled as epidemic spread.

(2f) **Escort** Rocha et al. [2010]: A sexual contact network between escorts and sex buyers in which STDs may be spread collected over six years. The size falls in the population definitions of urban area in most U.S. state that the health workers may deploy plans on.

(2g) **Epinion** Leskovec & Krevl [2014]: A trust network of a general consumer review site. This dataset is adopted mainly to show the scalability of the algorithms.

The results are shown in Table 3.3. We begin with initial observations and provide a more detailed analysis in the following section. In most cases, although the baselines behave differently for each data set, both versions of REMEDY make substantial improvements over them, and, as expected, Full-REMEDY exhibits better performance than Fast-REMEDY. In **Irvine**, the largest network for which all the algorithms are able to complete running within a 24-hour period, Fast-REMEDY and Full-REMEDY outperformed MaxDegree, the next best competitor, by 16.29% and 36.23% respectively. Fast-REMEDY also outperformed its next best competitor (MaxDegree) on **Epinion**, the largest network, by 37.44%. We further examined the performance of REMEDY for a range of β and γ values (see Fig. 3.5). FAST- and FULL-REMEDY continue to perform better than their closest competitors.

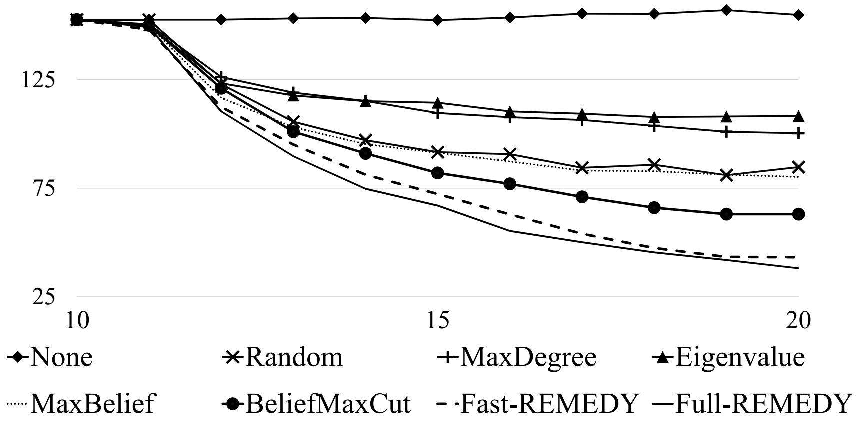
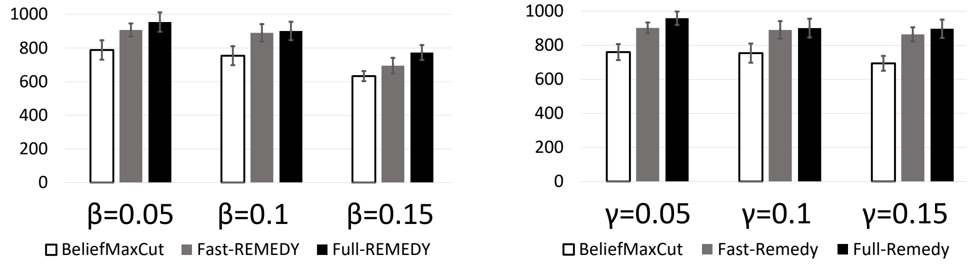


Figure 3.4: The average number of infected nodes (y-axis) vs. time (x-axis) of the ACTS stage in the **India** network.

Specifically, Fig. 3.4 shows the average number of infected nodes in each round on the **India** network. The values shown in Table 3.3 are the accumulation of the difference between None and each algorithm. Full-REMEDY steadily outperforms the other algorithms in each round and keeps decreasing the infected node number and Fast-REMEDY follows slightly behind it. The other algorithms, however, reach steady state and stop decreasing earlier.

Fig. 3.6 gives the running time of all the algorithms in different networks, sorted by size. Fast-REMEDY is about two orders of magnitude faster compared to Full-REMEDY, and takes about two hours on the largest network. All the algorithms that select a fixed set of nodes (Eigenvalue and



(a) γ=0.1

(b) β=0.1

Figure 3.5: Improvement over **None** (y-axis) under different parameter settings for **India** network.

MaxDegree) in every round are timed for the first round only for fairness of comparison. On **Escort** and **Epinion**, Eigenvalue, BeliefMaxCut and Full-REMEDY exceed 24 hours of computation time. It appears that only algorithms with complexity of $O(|V|^2)$ or less terminate within the time limit.

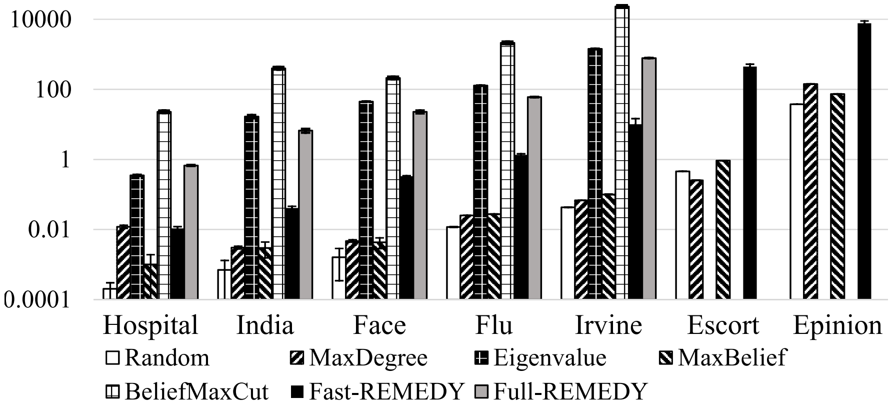


Figure 3.6: Computation time (y-axis, in seconds) in different contact networks in logarithmic scale. Note that the **Escort** and **Epinion** is significantly larger than the **Irvine** network and Fast-REMEDY still terminates under 10 minutes and 2 hours respectively.

3.3.2 Policy Analysis: Cure, Prevent, and Miss

To analyze the performance differences between algorithms, we introduce metrics that decompose the effect of active screening. As Figure 3.7 shows, screening a node has one of the following three effects:

- **Cure:** screening a node that is currently in the I state causes that node to transition to the S state.
- **Prevent:** screening a node in the S state prevents that node from entering the I state if it would have otherwise.
- **Miss:** screening a node in the S state has no effect if that node would not have transitioned to the I state.

We analyze the performance of the algorithms by measuring how many of their active screening actions result in each effect. Figure 3.8 shows the frequency of each effect for each algorithm, averaged over the rounds in the ACTS stage.

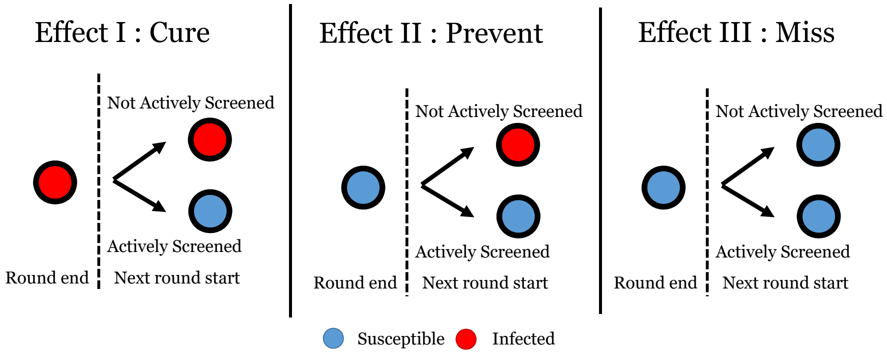
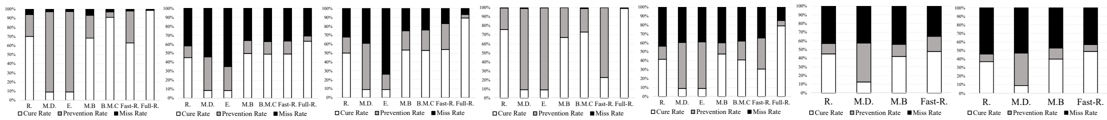


Figure 3.7: Three possible effects of active screening.

Active screening actions that cure infected nodes or prevent susceptible nodes from becoming infected will generally decrease the amount of infection in the network, whereas a miss has no effect. Therefore, we expect that algorithms with higher combined cure and prevention rates should have



(a) Hospital **(b)** India **(c)** Face-to-face **(d)** Flu **(e)** Irvine **(f)** Escort **(g)** Epinion

Figure 3.8: Frequency of each active screening effect, averaged over the ACTS stage. (R.: Random; M.D.: MaxDegree; E.:Eigenvalue; M.B.: MaxBelief; B.M.C.: BeliefMaxCut; Fasr-R:Fast-REMEDY; Full-R.:Full-REMEDY)

higher performance. This is indeed the case as the height of combined cure and prevention in Figure 3.8 is strongly correlated to the performance in Table 3.3 with only a few exceptions. The success of Fast- and Full-REMEDY can be explained primarily along these lines. Due to space constraints, a more detail analysis of the performance variation of each algorithm on each network based on their properties and three active screening effects is provided in the supplemental material.

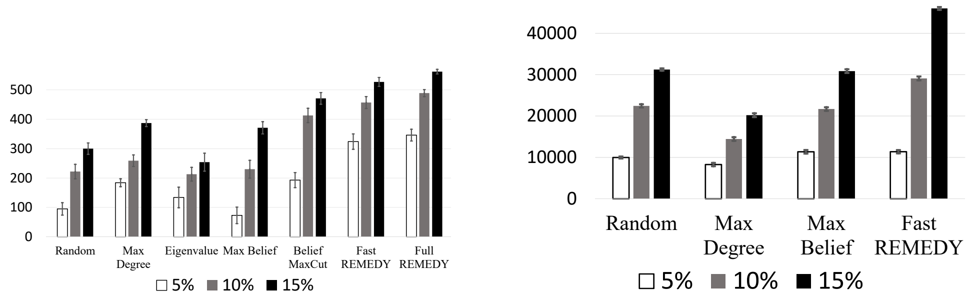
3.4 Deployment Considerations

In this section, we consider the effect of practical constraints on deployment of REMEDY in real-world active screening situations. Our ultimate aim is to be able to use the algorithms in settings such as active screening for tuberculosis in India.

Active screening implementations usually choose to screen whole districts where disease infections are severe. Recent experimental attempts that involved screening every first degree contact of reported patients (these are designated as *Naturally Cured* in this chapter)Holmes et al. [2017]. To address the challenge of deploying this work as a next step in active screening implementations, it is essential to take into account realistic barriers such as limited budget or missing information about the contact structure.

3.4.1 Impact of Budget

Determining the improvement an intervention can achieve with various budgets is critical when informing health policy. We therefore study the improvement possible over different budget values for two realistically modeled diseases: Influenza and Syphilis.



(a) Influenza

(b) STD

Figure 3.9: Improvement over **None** (y-axis) with specific disease parameters under different budget constraints (corresponding to 5%, 10%, 15% of total population).

Influenza. For Influenza, we use parameters estimated by previous literature through a continuous survey administered in a student residence hall community Dong et al. [2012]. The transition rate is estimated to be $\beta = 0.024$ and the self-cure rate is estimated to be $\gamma = 0.3$. We test the algorithms on the **Face-to-face** network, since this network is used to study the dynamics of SIS-type epidemic spread in its original paper Isella et al. [2011].

Fig. 3.9 (a) shows that both Fast-REMEDY and Full-REMEDY outperform other baselines under realistic settings. The difference grows larger as the budget increases. According to Prakash et al. [2012], such a network requires at least $k/|V| \geq \beta\lambda_A = 57\%$ for random screening to fully eradicate the disease. However, the epidemic dies out at the end of the 20th round (in all runs) when Full-REMEDY is deployed with a budget of only $k/|V| = 15\%$.

Syphilis. We use the syphilis parameters derived by Saad-Roy et al. [2016]. The natural cure rate is estimated to be $\gamma = 0.01$ and transmission rate $\beta = 0.035$. The network is the **Escort** network with

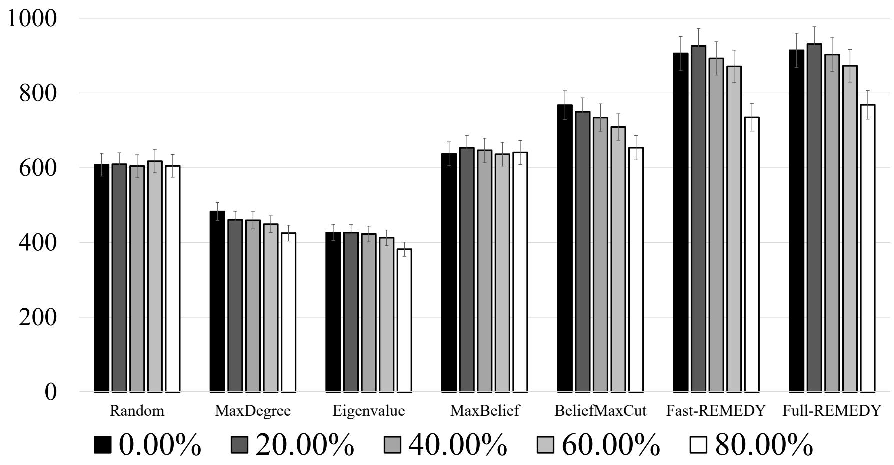


Figure 3.10: The improvement over **None** for different percentage of edge information lost in the **India** network.

16730 nodes, an STD contact network. Because the network is large, we show only the algorithms that do not exceed running time due to time complexity, which are Random, MaxDegree, MaxBelief and Fast-REMEDY. Fig. 3.9 (b) shows that Fast-REMEDY achieves significantly better results than all other baselines. On average, it saves 1140, 2900, and 4600 people from becoming infected every six months for 5%, 10% and 15% budgets, respectively.

3.4.2 Impact of Structure Uncertainty

In realistic settings, it is quite possible that the contact network is not known precisely. To simulate this, we randomly remove edges from the graph and then provide the graph with missing edges as input to the algorithms. All the algorithms make decisions based on this graph with missing edges without knowing such fact while the disease spread happens along the true network with all edges.

Both version of REMEDY still significantly outperform other baselines even when the percentage of edges randomly removed is as high as 80% (Fig. 3.10). In other words, it is able to outperform the other implementations with only 20% of the contacts are known.

3.5 Summary

This chapter presents the REMEDY agent for a novel active screening multiagent problem (ACTS) that takes into account real world constraints such as uncertain health states and limited intervention resources. No previous work has addressed the challenge of such uncertainty raised from the emerging application active screening of recurrent diseases. Active screening provides a powerful yet expensive means to control disease spread in the public health domain that passive screening cannot achieve due to its latency of cure. The agent is developed to assist our collaborator in India to decide who and when health workers should invest their limited resources and improve the current practice approach. We introduced two variant of algorithms the agent used, Full-REMEDY and Fast-REMEDY and examined them on various real human contact networks and realistic disease parameters to show their superior performance over any past approach.

4

Active Screening Problem: Long-term Planning Using Reinforcement Learning

4.1 Problem Formulation

In this chapter, we focus on a sequential decision making problem with a large time horizon, where in each round (time step) we aim to optimally select which individuals in a social network to actively screen, so that the expected number of un-infected individuals over the time horizon is maximized.

4.1.1 Markov Decision Process Formulation

We start by formulating the active screening problem as a MDP. **States:** The hidden state of our problem is the combinatorial health state of each individual node which is partially observable. To represent the observation uncertainty in the current state, we follow chapter 3 by defining a *belief state* b_v for every node v, which can be interpreted as the approximate probabilities of each node being infected.

Actions: Given the current state, the agent can choose any subset of nodes $C \subseteq V, |C| \leq k$ to screen. The size of the action space is $\binom{V \backslash \mathbf{o}_t}{k}$ at each round, where $\mathbf{o}_t$ is the set of nodes that are passively screened (so there is no advantage in actively screening them).

Rewards: The objective of the active screening problem is to maximize the accumulated number of susceptible nodes. It is natural to consider the step wise reward signal as number of susceptible nodes after the active screening, denoted as $r_t = \Sigma_{v \in V} 1_{\mathbf{x}_t^v = S}$. Since every infected individual has a fixed probability γ of being observed by the agent, the step wise reward can be easily estimated.

Transitions: In belief of the health states, screened or observed nodes ($v \in \mathbf{o}_t \cup \mathbf{a}_t$) are updated by their ground truth values and the remaining nodes are updated by inferring their posterior probabilities. The key to state transition is the update of belief state, which is defined following chapter 2.

4.2 Methodology

Despite a well defined MDP, it is extremely challenging to solve it due to various reasons. One of the main challenges is that both state space and action space we are facing are high-dimensional. For the state space, even when the true state is available, there are a total of $2^{|V|}$ possible states. When uncertainty is involved, the state values are continuous and therefore the number of states is infinitely large. Furthermore, the states of individual nodes are not independent from each other, but are correlated due to potential contacts from the network. For the action space, in each time period we need to choose a combination (subset) of nodes from the entire network. For a reasonably large network, this combinatorial action space grows exponentially with respect to the screening budget k, and becomes intractable as k typically scales as the graph size grows. In the following we show how these challenges are handled in our approach. A summary of notations related to the algorithm is included in Table 4.1.

Table 4.1: Notations used in chapter. Most of them are consist with chapter 2 except the newly introduced symbols.

Symbol	Description
G	contact network
S	susceptible state
I	infectious state
β	transmission probability
γ	cure probability
t	time step
T	time horizon
k	screening budget for each time step
$\mathbf{x}_t$	true state at time t (not available in testing)
$\mathbf{a}_t$	set of nodes actively screened (action at time t)
$\mathbf{o}_t$	set of self-report nodes (observation at time t)
$\mathbf{b}_t$	Belief state at time t
$\mathbf{s}_t$	state representation for Q function
r_t	step wise reward
Q	Q function
α	future discount factor
τ	auxiliary coefficient for curriculum learning
$\bar{r}_t$	Initial step wise reward for curriculum learning

4.2.1 Basics of DQN

Due to the superior performance of RL algorithms in solving large scale MDPs Mnih et al. [2015], Silver et al. [2016, 2017], we adopt RL as the basis of our solution. More specifically, the backbone of our approach is a hierarchical RL algorithm based on DQN. In this sub-section, we first introduce the basics of RL and DQN, following which we then describe several ideas that further adapt DQN to our formulated problem. We need to emphasize that we do not claim novelty in each of the adaptions, but instead the novelty lies in the innovative way of combining the ideas into solving the particular problem of interest.

RL is a learning framework where agents learn to perform actions in an environment so as to maximize a certain objective. The two underlying components of RL are the environment, which is defined as the MDP in this chapter, and the agent, which represents the learning algorithm. At each time step t, the agent takes an *action* based on its *policy* $\pi(\mathbf{a}_t|\mathbf{s}_t)$, where $\mathbf{s}_t$ and $\mathbf{a}_t$ are respectively the state and action of the MDP defined above. The agent then interacts with the environment with the selected action and the environment returns a reward r_t for that action as well as the state $\mathbf{s}_{t+1}$ of the next time step. Q-learning Watkins & Dayan [1992] is a value-based RL approach that is based on the notion of Q-function (i.e., state-action value function). The Q-function measures the expectation of accumulated rewards of an action $\mathbf{a}_t$ given state $\mathbf{s}_t$. In the training phase of Q-learning, the policy usually exploits the action with the highest Q-value with a high probability $1 - \varepsilon$, and explores random actions with a small probability ε. The Q-function is typically estimated using the Bellman equation: $Q^{i+1}(\mathbf{s}_t, \mathbf{a}_t) = r_t + \alpha \max_{\mathbf{a}_{t+1}} Q^i(\mathbf{s}_{t+1}, \mathbf{a}_{t+1})$, where i indicates the training iteration and α is the discount factor. DQN Mnih et al. [2013] improves Q-learning by representing it using deep neural networks, together with other techniques like experience replay over a number of episodes ($\mathcal{E}$), which basically stores the historical training trajectories in a "replay buffer" and updates the Q-function by minimizing the loss function $(y - Q(s, a))^2$ with batch data from the replay buffer using gradient descent algorithms. Here y is the "target" which is estimated using the above Bellman equation (a technique usually called Temporal Difference learning), and $Q(s, a)$ is directly obtained by feeding s and a to the Q-function.

4.2.2 GCN-based Function Approximator

With the deep neural networks based function approximator, DQN addresses the exponentially large and continuous state space in our formulated MDP. However, one shortcoming of such a function approximator is that it does not capture the intrinsic correlations between node features. Intuitively, the infectious statuses of linked nodes in a social network are inter-dependent.

Graph convolutional neural networks (GCNs) Kipf & Welling [2017] embed the graph structure itself into its network directly, and thus have superior performance on graph type inputs. Each layer of GCN is given by $\mathbf{z}^{l+1} = \sigma(\mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{\frac{1}{2}}\mathbf{z}^{l}\mathbf{W})$, in which $\mathbf{z}^{l}$ is the input of l-th layer, $\mathbf{D}$ is the diagonal node-degree matrix that normalizes the adjacency matrix $\mathbf{A}$, $\mathbf{W}$ is the trainable weight matrix and $\sigma(\cdot)$ is the activation function. The convolutional layers in GCNs can facilitate the nature of message passing and automatically aggregate the information from neighboring nodes. Such message passing is similar to the infection spread in our epidemic model. The advantage of using GCNs is that we do not need the hand-crafted graph features to represent the information about the graph structure such as the node degrees or eigenvalue of adjacency matrix Li et al. [2012]. Inspired by recent advances that combine the power of RL and GCNs-based deep function approximators Khalil et al. [2017], Qiu et al. [2019], Kamarthi et al. [2020], we use GCNs in this chapter to represent the Q-function.

Our adaptation of the GCNs takes the belief state as input. The action and observation can be naturally encoded in the belief state as we can update the corresponding elements in the belief state vector to their true state. Thus we do not need to encode them as additional features.

We thus combine the observation with the graph structure that is represented as the adjacency matrix $\mathbf{A}$ to our state representation in a very structured way. The GCNs learn the underlying graph embedding and automatically form the representation and output the Q-value estimation for our RL agent.

4.2.3 Sequence of Sequence Framework

In addition to the challenges that arise from the high-dimensional and graph-structured state space, as described previously, another challenge is the combinatorial action space in each time step of screening. To address this challenge, we propose a hierarchical RL approach by re-formulating each time step in the original MDP as a sub-sequence of decisions by itself. We call this framework sequence of sequence (SOS). We refer to the original multi-rounds of active screening as *time* sequence. Correspondingly, we refer to the sub-sequence problem of selecting k nodes in each round as the *budget* sequence. In each of the budget sequence, we are solving a separate sequential decision making problem with a final reward R_t, a finite time horizon of k, and an action space $V \setminus \mathbf{o}_t$ whose size is equal to the network size.

The SOS framework allows for a tractable action space which can be used by an RL algorithm. However, two additional issues arise in such conversion of the action space. First, although the states and actions are well represented by GCNs, the algorithm does not take into account the remaining budget at the budget sequence. In fact, the policies should be very different when there is plenty of budget left versus little budget left. Intuitively, this is because the actions taken when there is more budget left should consider more about its future effect, while actions taken when there is less budget left tend to be more myopic. Therefore, a single-agent framework in the budget sequence, which treats all states equally for different remaining budget, usually does not work well.

Second, by introducing the SOS framework, only the final step in the k budget-steps for the budget sequence gets a reward signal. The sparseness of reward is known to slow the convergence of RL Irpan [2018]. Therefore distributing the reward R_t to each action in the budget sequence for proper reward signaling is a non-trivial task.

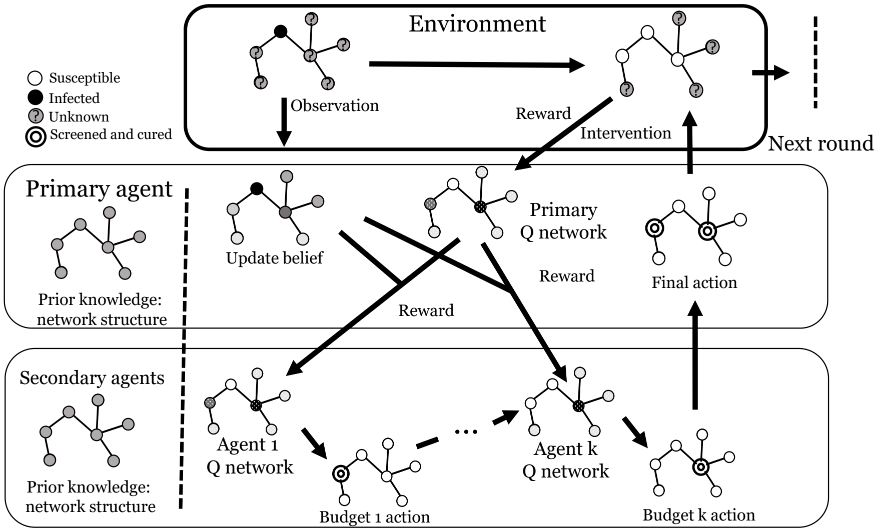


Figure 4.1: The overall flow of one round decision making in a budget sequence using our two-level RL structure. The top row depicts the environment, the second row is the workflow of the primary agent, and the third row is the workflow of the k secondary agents. The primary agent starts by observing the initial state of the environment. It then updates the belief of the state. The belief state is passed down to the first secondary agent, which then evaluates the value of each feasible node action using its own Q network and decides which node to select. The successive secondary agents work sequentially until the k-th secondary agent, i.e., when budget k is spent. The selected actions are collected to get the final action set $\mathbf{a}_t^I$ which is uploaded to the primary agent. The primary agent then interacts with the environment using this action and gets reward signals.

4.2.4 Primary and Secondary Agents

Inspired by hierarchical RL Dayan & Hinton [1993], Parr & Russell [1998], Sutton et al. [1999], Dietterich [2000], we propose a multi-agent RL approach with a two-level structure that manages the time and budget sequences in a hierarchical way. The overall flow of the two-level structure is depicted in Fig. 4.1. The idea is to capture the remaining budget information by having k secondary agents, where each secondary agent maintains a policy for a different remaining budget value. The primary agent, as shown in Algorithm 4, manages the time period reward signal in a given time step of the time sequence. The reward signal acts as the secondary agents' total reward in the budget sequence and is distributed over the k secondary agents. In the following, we use superscripts I and II to distinguish concepts that correspond to the primary and secondary agents.

Primary Agent In the time sequence, the primary agent works as follows. First, it receives the combinatorial action and the secondary agents' memories that store the trajectories from the secondary agents (line 6). Passing this action to the environment, it receives the observation and reward (line 7). It then handles the new state representation of the next time step (lines 9 and 10). Finally, it updates both primary and secondary agents' memories ($\mathcal{M}^I$ and $\mathcal{M}^{II}$) (lines 11 and 12). Each element of $\mathcal{M}^I$ and $\mathcal{M}^{II}$ is basically a tuple of state, action, next state and rewards that will be used to train the primary and secondary agents' Q-functions. Note that the state of the secondary agents is slightly different from that of the primary agent, which will be explained in the next paragraph. These memories are used to update the GCN-based Q-functions by minimizing the loss function $(y - Q(s, a))^2$ through gradient descent (lines 16 and 17), where y is the target. At each time step t, the target of each agent is given by:

$$y^I = r^I_t + \alpha Q^I(\mathbf{s}^I_{t+1}, \mathbf{a}^I_{t+1}), \tag{4.1}$$

$$y^{II}_i = r^{II}_i + \alpha Q^{II}_{i+1}(\mathbf{s}^{II}_i, a^{II}_{i+1}) \text{ for } i = 1, \ldots, k-1, \tag{4.2}$$

$$y^{II}_k = r^{II}_k + \alpha Q^I(\mathbf{s}^I_{t+1}, a^I_{t+1}). \tag{4.3}$$

Note that in lines 5 and 8, the belief state and reward are obtained using the idea of curriculum learning that mitigates the reward sparseness issue for the secondary agents. We will elaborate this idea in the next subsection.

Secondary Agents As for the budget sequence, instead of training a single agent and performing a batch selection of nodes, we train k secondary agents to handle each budget sequentially. As described above, the purpose of doing so is to differentiate secondary agents who know there is plenty of budget left and those who know there is little budget left, so they could learn different policies. The state $\mathbf{s}^{II}_i$ of a secondary agent i is obtained by encoding the primary agent's action (i.e., the set of nodes selected so far) taken upon the state of the previous budget step. The action a^{II}_i for each secondary agent i is to

Algorithm 4 Primary Agent

```
1: for $episodes = 1, ..., \mathcal{E}$ do
2:     Initialize and acquire initial belief $(\mathbf{b}_0)$ and observation $(\mathbf{o}_0)$
3:     $\mathbf{s}^I_0 \leftarrow$ Graph Embeding$(\mathbf{A}, \mathbf{b}_0)$
4:     for $t \in 0, ..., T$ do
5:         $\tilde{\mathbf{b}}_t \leftarrow$ Curriculum Belief Transform$(\tau, \mathbf{b}_t)$
6:         $\mathbf{a}^I_t, m^{II} \leftarrow$ Secondary Agent$(\tilde{\mathbf{b}}_t, Q^I)$
7:         $\mathbf{o}_{t+1}, r^I_t \leftarrow$ Environment$(\mathbf{a}^I_t)$
8:         $\tilde{r}^I_t \leftarrow$ Curriculum Reward Transform$(\tau, r^I_t)$
9:         $\mathbf{b}_{t+1} \leftarrow$ Belief Update$(\mathbf{b}_t, \mathbf{o}_t, \mathbf{a}^I_t)$
10:        $\mathbf{s}^I_{t+1} \leftarrow$ Graph Embeding$(\mathbf{A}, \mathbf{b}_{t+1})$
11:        $\mathcal{M}^I \leftarrow \mathcal{M}^I \cup \left\{(\mathbf{s}^I_t, \mathbf{a}^I_t, \tilde{r}^I_t, \mathbf{s}^I_{t+1})\right\}$
12:        $\mathcal{M}^{II} \leftarrow \mathcal{M}^{II} \cup m^{II}$
13:    end for
14:    Decrease $\tau$
15: end for
16: Fit $Q^I$with regressor net using $\mathcal{M}^I$
17: Fit $Q^{II}_0 \ldots Q^{II}_{k-1}$with regressor nets using corresponding $\mathcal{M}^{II}$
```

choose one node to add to the primary agent's action set $\mathbf{a}^I$. $\mathbf{a}^I$ is initialized as an empty set (in line 1) and will be updated by appending actions from each secondary agent. In the for loop that represents a budget sequence (from line 3 to line 8), each secondary agent i will select the action a^{II}_i that maximizes its Q-function $Q^{II}_i(s^{II}_i, a^{II}_i)$ and add it to the primary agent's action set $\mathbf{a}^I$ (lines 4 and 5). After that, it receives the reward in line 6, which is obtained from the primary agent (as a proxy) in a temporal difference learning manner. Next, the secondary agent will encode the primary agent's action $\mathbf{a}^I$ (i.e., the set of nodes selected so far) into its current state $\mathbf{s}^{II}_i$, which is used as next state $\mathbf{s}^{II}_{i+1}$ and pass this information to the next secondary agent (line 7). Finally, it stores the above information as memory, so it can be used later to update the Q-functions in Equations (4.1)- (4.3). For extremely large graphs, we reduce the memory and computation time by assigning fewer than k secondary agents, where each secondary agent is responsible for a portion of the budget instead. For example, for a budget of 20, if we assign 10 secondary agents, each secondary agent needs to select $20/10 = 2$ nodes at a time.

Algorithm 5 Secondary Agents

```
$\mathbf{a}^I, m_1^{II} ... m_k^{II} \leftarrow \emptyset$
$\mathbf{s}_0^{II} \leftarrow Encoding(\mathbf{s}^I, \mathbf{a}^I)$
**for** $i \in 1, ..., k$ **do**
    $a_i^{II} \leftarrow \arg\max Q_i^{II}(s_i^{II}, a_i^{II})$
    $\mathbf{a}^I \leftarrow \mathbf{a}^I \cup a_i^{II}$
    $\mathbf{r}_i^{II} \leftarrow Q^I(\mathbf{s}^I, \mathbf{a}^I) - Q^I(\mathbf{s}^I, \mathbf{a}^I \setminus a_i^{II})$
    $\mathbf{s}_{i+1}^{II} \leftarrow Encoding(\mathbf{s}^I, \mathbf{a}^I)$
    $m_i^{II} \leftarrow m_i^{II} \cup \left\{ (\mathbf{s}_i^{II}, a_i^{II}, \mathbf{r}_i^{II}, \mathbf{s}_{i+1}^{II}) \right\}$
**end for**
**return** $\mathbf{a}^I, m^{II}$
```

4.2.5 Curriculum Learning

As discussed earlier, a major issue with the two-level framework is the sparseness of rewards for the secondary agents. Inspired by curriculum learning Bengio et al. [2009], we address this by incrementally increasing the complexity of the learning tasks for the secondary agents. This is called *Curriculum Transformation* in lines 5 and 8 in Algorithm 4 and is described as the following equations:

$$\tilde{\mathbf{b}}_t = \tau \mathbf{x}_t + (1-\tau)\mathbf{b}_t, \tag{4.4}$$

$$\tilde{r}_t^I = \tau \bar{r}_t + (1-\tau) r_t^I, \tag{4.5}$$

where $\tilde{\mathbf{b}}_t$ and $\tilde{\mathbf{r}}_t^I$ are respectively the belief state and reward of the primary agent (used to update the target values for both the primary and secondary agents) after curriculum transformation. τ is an auxiliary coefficient that gradually decreases from 1 to 0 in the first few epochs of training. It adjusts task difficulties from a relatively easier problem ($\tau = 1$) to the original problem ($\tau = 0$). At the early stage of training ($\tau = 1$), we warm up the learning by feeding the algorithm with the true state information $\mathbf{x}_t$ (Eq. (4.4)). Moreover (Eq. (4.5)), we set the reward to be $\bar{r}_t = \sum_{v \in \mathbf{a}_t^I} 1_{b_t^v = S}$, which means the total number of infected nodes in the action set $\mathbf{a}_t^I$, instead of in the total number of susceptible nodes S.

In this way, the algorithm learns to greedily cure nodes that are infected. As the training continues, decreasing the auxiliary coefficient τ takes two effects. First, it gradually removes the true state information. It is worth noting that the true state is only used in warming up the training, and is not used during testing. Second, it shifts the reward from being constrained in the set of infected nodes to the true reward, and explores potentially more optimal actions outside the set of infected nodes. When τ is 0, the belief and reward become identical to the original problem ($\tilde{\mathbf{b}}_t = \mathbf{b}_t$ and $\tilde{r}_t^I = r_t^I$).

An alternative is to train a single agent and encode the remaining budget information directly as part of the state. However, we show via ablation study in figure 4.2 that this leads to sub-optimal solution quality. To show the effectiveness of our two-level RL framework and the curriculum learning component, we conduct an ablation study on a sample network Face-to-face. Fig. 4.2 shows the ablation study results on the Face-to-face network. We evaluate 4 settings: (i) Single agent without curriculum learning (CL); (ii) Two-level (k agents) RL without CL; (iii) Single agent with CL and (iv) Full (i.e., two-level RL with CL). Note that in (i) and (iii), the remaining budget is directly hard-coded as part of the state information to the GCN. By comparing (i) with (iii) or comparing (ii) with (iv), we can see that curriculum learning is critical in improving the solution quality. On the other hand, by comparing (i) with (ii) or comparing (iii) with (iv), we can see that the two-level primary and secondary agents framework, which trains a different secondary agent policy, is also improving the solution quality by a large margin. On the contrary, hard-coding the remaining budget into the state leads to sub-optimal solution quality.

4.3 Experiments

Datasets We evaluate the effectiveness of our proposed approach on different real-world contact networks used in chapter 2 for comparison.

Experimental setting In all experiments, we fix the passive screening rate γ to 0.05. Due the the

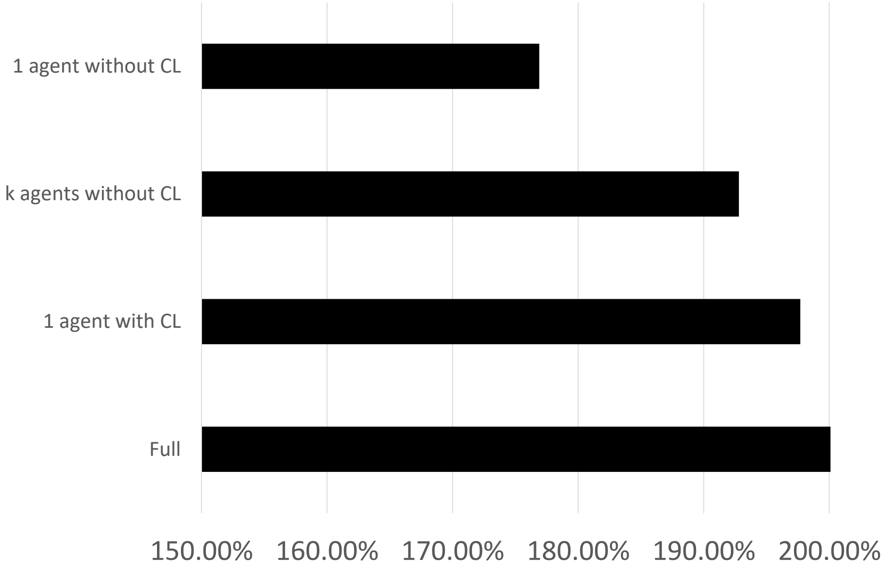


Figure 4.2: Ablation study run on the Face-to-face network. The x-axis is the percentage of improvement over no-intervention. The y-axis denotes different variants of our approach.

high diversity of the networks, it is difficult to find one fixed transmission rate β that is suitable for every network. A fixed transmission rate is either too high for the dense networks so that the whole network will become infected no matter what policy is deployed, or too low for the sparse networks so that the disease will be eradicated without any intervention. We thus adjust the transmission rate according to the density of the network. Specifically, we adjust β based on the spectral radius $1/\lambda_A^*$, also known as the epidemic threshold, where λ_A^* is the largest eigenvalue of the network adjacency matrix. If there are no additional interventions, the disease will not be eradicated eventually if and only if $\beta > \gamma/\lambda_A^*$ Wang et al. [2003]. We set $\beta = 10\gamma/\lambda_A^*$, which is 10 times the value corresponding to the epidemic threshold. We further set $k = 0.1|V|$ for the active screening budget in each time period. Finally, we set the total time horizon to $T = 100$. All the results presented are averaged over 30 trials. For our RL approach, we set the future discount factor α to 0.98, exploration probability in the epsilon greedy approach is 0.1 and the learning rate is 0.005. We trained the RL agents for 100 episodes for 100 iterations of refits. The memory capacity of the relay buffer is set to 5000 tuples for

Table 4.2: Average improvement of different algorithms. *Full-REMEDY* does not scale to $T = 100$. Thus its results are not included. The numbers in the brackets are improvements over the best alternative *Fast-REMEDY*.

Network	Reduction in infections compared with no intervention				
	Eigenvalue	Max-Degree	Random	Fast-REMEDY	**RL**
Hospital	1019±99	1015±131	2344±136	3837±340	**4196±416 (9.4%)**
India	2668±258	3115±292	6388±259	10033±518	**11270±501 (12.3%)**
Face-to-face	4225±211	4705±219	8948±173	9919±499	**13283±301 (33.9%)**
Flu	7706±190	7725±203	9636±163	10298±460	**11743±503 (14.0%)**
Irvine	48490±298	49163±378	42277±270	53159±673	**65128±781 (22.5%)**

each agent. We used the sigmoid activation function. There are four layers of sizes 8,16,8 and 32. For all graphs except the largest one, the training finishes within 3 hours on a laptop with 6 cores, 2.60 GHz intel CPU, and 16 GB RAM. For the largest graph, *Irvine* network, it takes about one day to finish on the same laptop and is significantly shortened after using an HPC.

Baselines We simply call our approach RL. The baselines we are testing against are (i) *Eigenvalue* (greedily choosing nodes that decrease the largest eigenvalue of the remaining sub-graph after removal until the budget is exhausted), (ii) *MaxDegree* (choosing *k* nodes with the largest degrees), (iii) *Random* (randomly selecting nodes), (iv) *Full-REMEDY* (the algorithm in chapter 2 that is un-scalable to large time horizons) and (v) *Fast-REMEDY* (the scalable version of *Full-REMEDY* that does not account for the future effect of actions).

4.3.1 Solution Quality

Table 4.2 shows the increase in average reward compared with no intervention. We can see that our approach outperforms the state-of-art (i.e., *Fast-REMEDY*) by a margin of $9 - 33\%$. There are a few observations worth noting. First, we are implementing the same baselines on the same datasets with significantly larger time horizon compared with chapter 2. The ranking of these baselines is consistent with chapter 2 that are run over a shorter time horizon of 10. Furthermore, results on *Hospital* and *Flu* show a more significant difference as we adjust the transmission rate according to

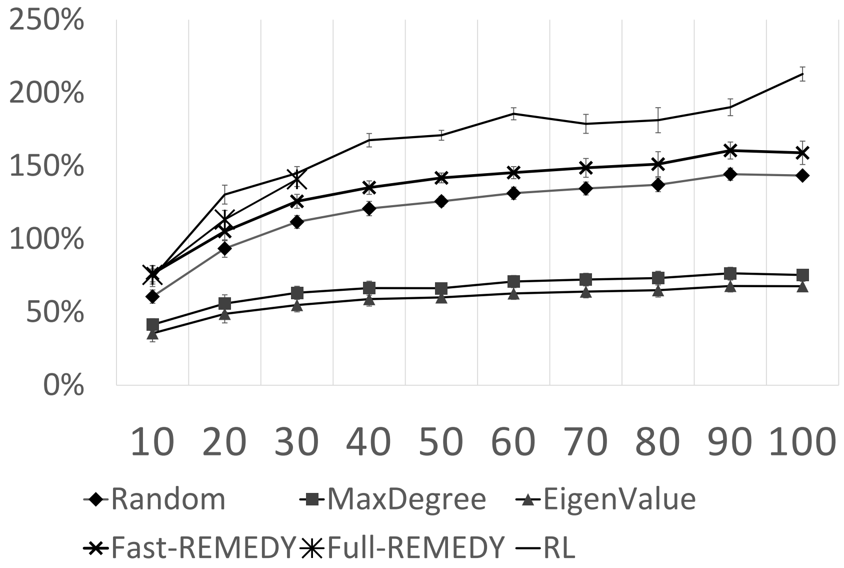


Figure 4.3: The performance of each algorithm for different time horizons in the Face-to-face network. The x-axis is the time horizon and the y-axis is the improvement of solution quality over no intervention.

the graph density. Second, *Eigenvalue* and *MaxDegree* baselines perform similarly to each other when we increase the time horizon where as in chapter 2, *MaxDegree* clearly outperforms *Eigenvalue* when the time horizon is short. This is expected as the *Eigenvalue* baseline is aimed for long term disease eradication by increasing the epidemic threshold and thus preforms better in the long term. Finally, in *Face-to-face* and *Irvine* networks, our approach performs significantly better compared with the best baseline *Fast-REMEDY*. Interestingly, these are also the networks where *Full-REMEDY* outperforms *Fast-REMEDY* in chapter 2. In these networks, the algorithms can benefit more by looking ahead compared with other networks.

4.3.2 Scalability Against Time Horizon

To study how well our approach scales against planning time horizon compared with baselines, we conduct experiments on various time horizons ranging from $10 - 100$. Figure 4.3 shows the performance on the face-to-face network for each algorithm on the y-axis when varying the time horizon on the x-axis. *Full-REMEDY* does not scale over time horizons longer than 30 even on a high performance computer and our approach is in par or better with its performance in these short time horizons. We pick face-to-face network as an example and similar trends can be observed for all the networks. Particularly, in the largest network Irvine, our approach scales 10 times of that in *Full-REMEDY*, meanwhile with better solution quality compared with *Fast-REMEDY*.

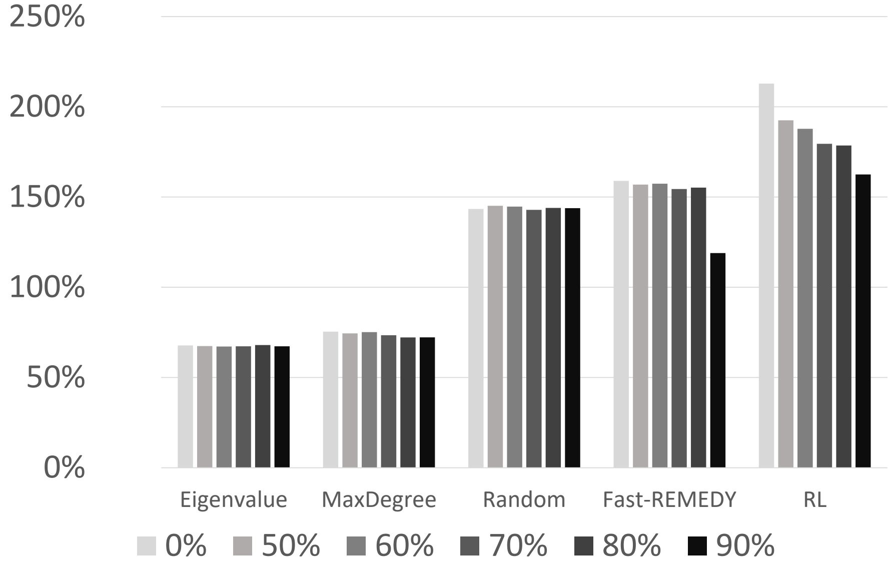


Figure 4.4: The performance of each baseline for different edge removal fractions. The x-axis indicates the improvement over no intervention and y-axis indicates the percentage of removed edges.

4.3.3 Robustness Against Structure Uncertainty

Although we assume perfect knowledge of graph structure (sans the infectious state of the individuals), one of the main obstacles in implementing active screening in practice is the lack of this perfect knowledge. To evaluate how robust different methods are against structural properties, we design and run the methods on two models for structural uncertainty. In the first one, we assume a constant fraction of the edges are unobserved where this fraction is a parameter. In the second one, we assume all the edges adjacent to a constant fraction of nodes in the graph are unobserved. In both of these

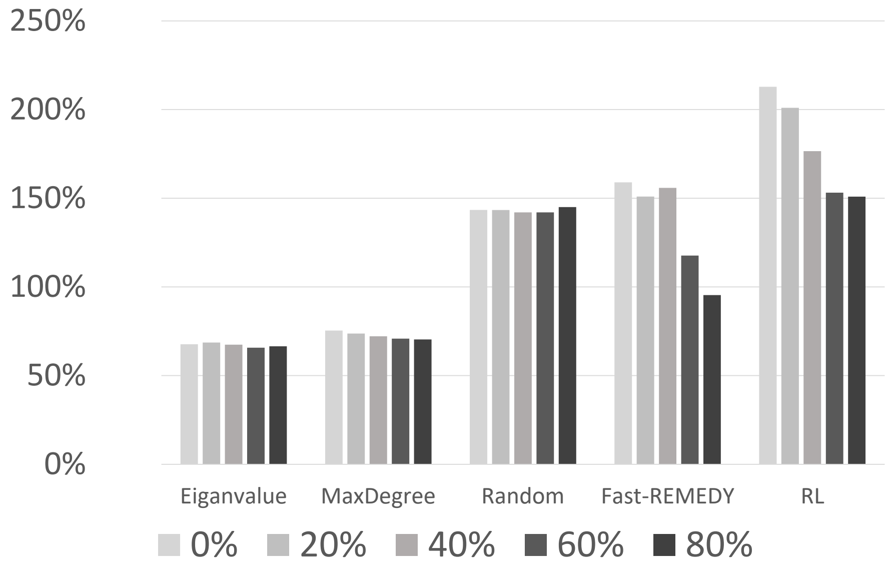


Figure 4.5: The performance of each baseline for different node removal fractions. The x-axis indicates the improvement over no intervention and y-axis indicates the percentage of nodes whose adjacent edges are removed.

models, we train our RL policy on the observed network and measure the performance of the learned policy on the actual network. We call these models edge and node removal, respectively.

We point out that in the first model some of the properties of the original graph like the nodes with maximum degree are preserved DuBois et al. [2012]. In the second model, many of the properties of the original network like centrality are likely to be changed Smith & Moody [2013].

Although our approach is the only learning algorithm that can benefit from different training sub-graphs, to make fair comparison, we train our RL policy on a single sub-graph. This is corresponding to the real world scenario where partial contact information is missing without being noticed. The results are summarized in Figures 4.4 and 4.5, respectively. Although the performance of our approach decays as the uncertainty increases, it still outperforms all the other baselines. Again, we show the result of face-to-face network as an example due to space limit. The results of the other networks have similar trends and can be found in figure 4.8 to figure 4.11.

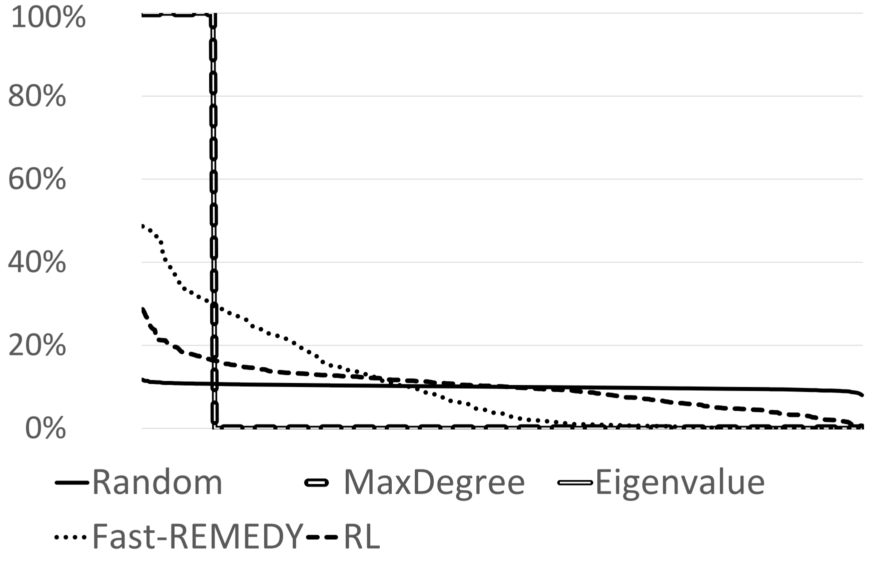
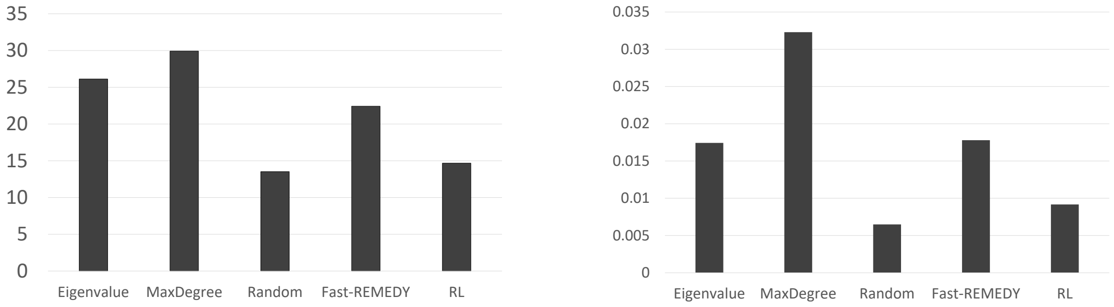


Figure 4.6: The node picking frequency distribution for each algorithm, sorted from high to low.

4.3.4 Policy Analysis

To gain insight on the patterns of how different approaches select nodes, we study the frequency in which each of the nodes is selected. The results are summarized in Figure 4.6 for the Face-to-face network. Similarly, results for other networks are in figure 4.9. Each point in x-axis represents a node, sorted by the frequency of being picked by the corresponding algorithm. The y-axis represents the frequency a certain node is picked by the algorithm. We sort all the algorithm's node picking frequency in order to show their distribution. In this figure, *Random* is the fairest algorithm as it picks each node with equal frequency, whereas *MaxDegree* and *Eigenvalue* always pick the same set of nodes as we have a static network. *Fast-REMEDY* selects the most frequently picked nodes half of the times while almost never picks 40% of the nodes. Our RL approach does not pick the structurally important nodes as often as *Fast-REMEDY* does, which is shown in figure 4.7. It is less structure dependent and tends to select a larger variety of nodes. By taking future actions into account, it depends more on the observation information and has a surprising side effect that outputs a fairer policy that gives more nodes chances to be screened.



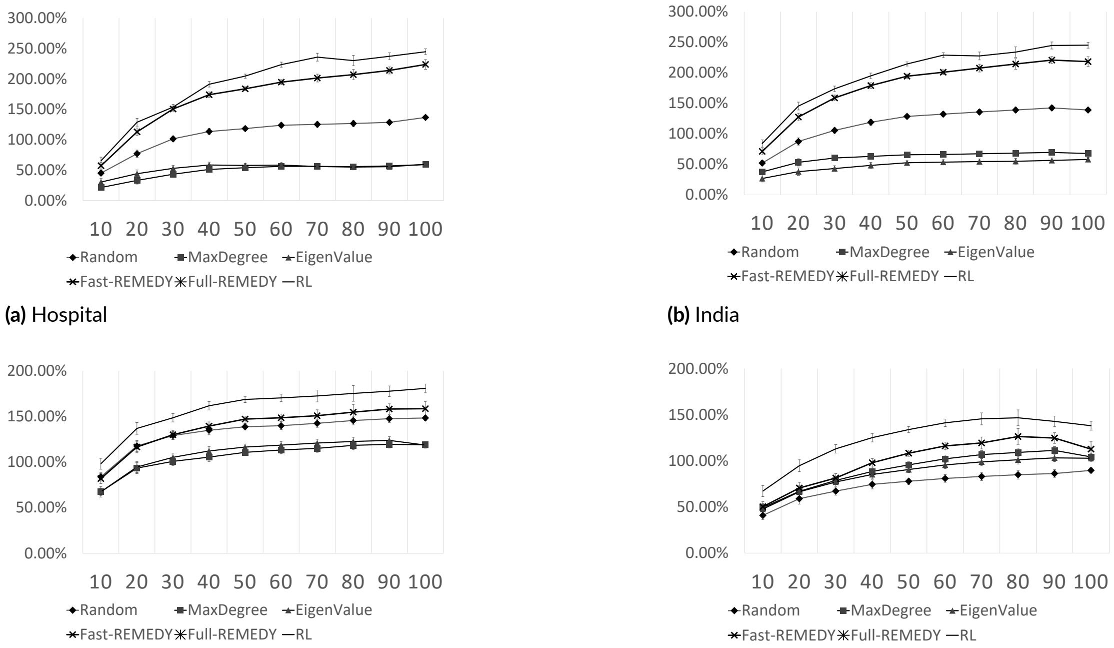
(a) Average Degree

(b) Average Betweeness

Figure 4.7: Average degree & betweeness of the nodes picked.

4.4 Summary

This chapter makes the first attempt at addressing the multi-round active screening problem using reinforcement learning. I formulate the problem as a MDP with high dimensional state and action spaces, which cannot be efficiently solved using classical RL algorithms like DQN. I then design several innovative adaptations to vanilla DQN, including GCN-based value function approximator that exploits the correlations of nodes, a primary-secondary agents framework that decomposes the combinatorial action selection in each time period into a sub-sequence of node selection, and a curriculum learning component that addresses the sparseness of reward for the secondary agents. Empirical results show that in terms of solution quality, the RL approach outperforms *Fast-REMEDY* by a margin of $9\% - 33\%$, and works better than baselines even with network structure uncertainty. In the largest network we experimented which 1899 nodes, our approach is able to scale up to a planning horizon 10 times that in state-of-the-art approach *Full-REMEDY*. Interestingly, policy analysis results show that compared with most baselines (except for *Random*), the RL approach is fairer in the sense that it tends to spread the screening across different nodes. For future work, we plan to incorporate uncertainty on the graph structure in training our RL algorithms and further improve the robustness of this approach.



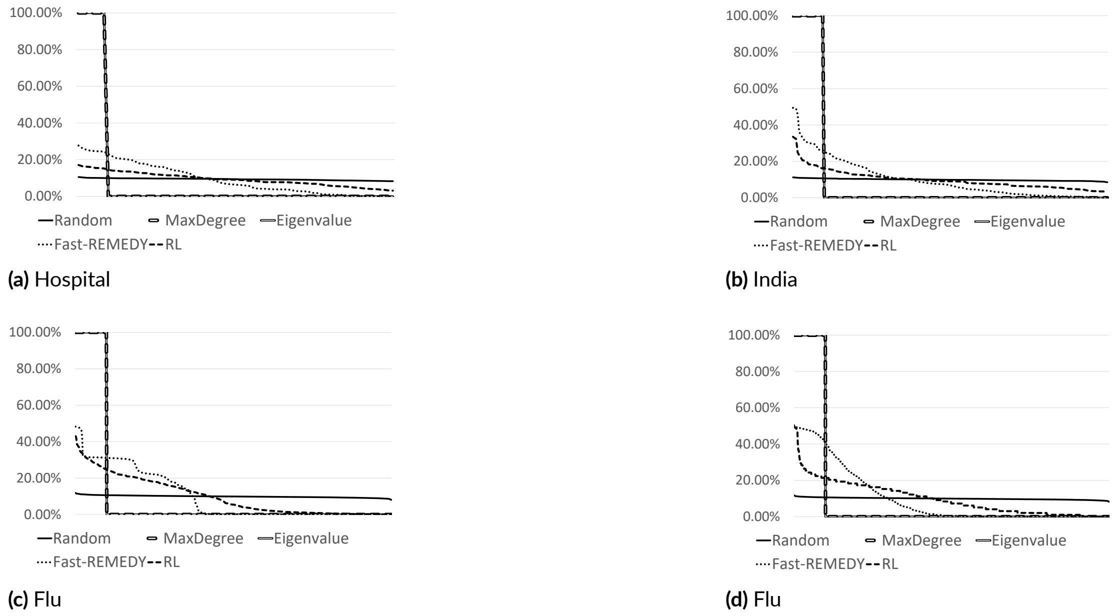
(a) Hospital

(b) India

(c) Flu

(d) Irvine

Figure 4.8: Performance under node information removal.



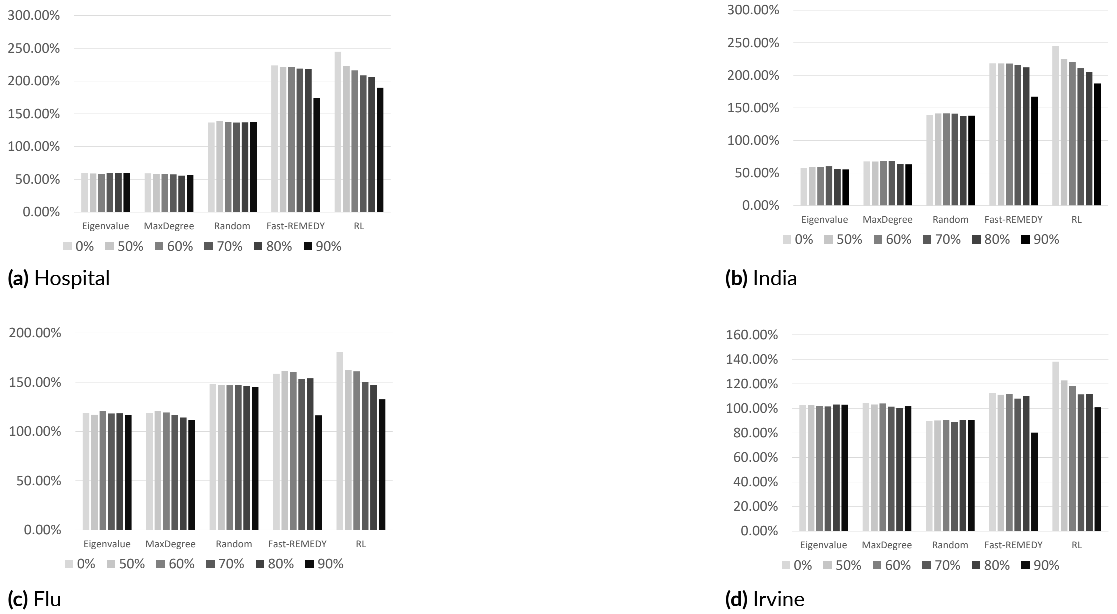
(a) Hospital

(b) India

(c) Flu

(d) Flu

Figure 4.9: Node picking frequency.



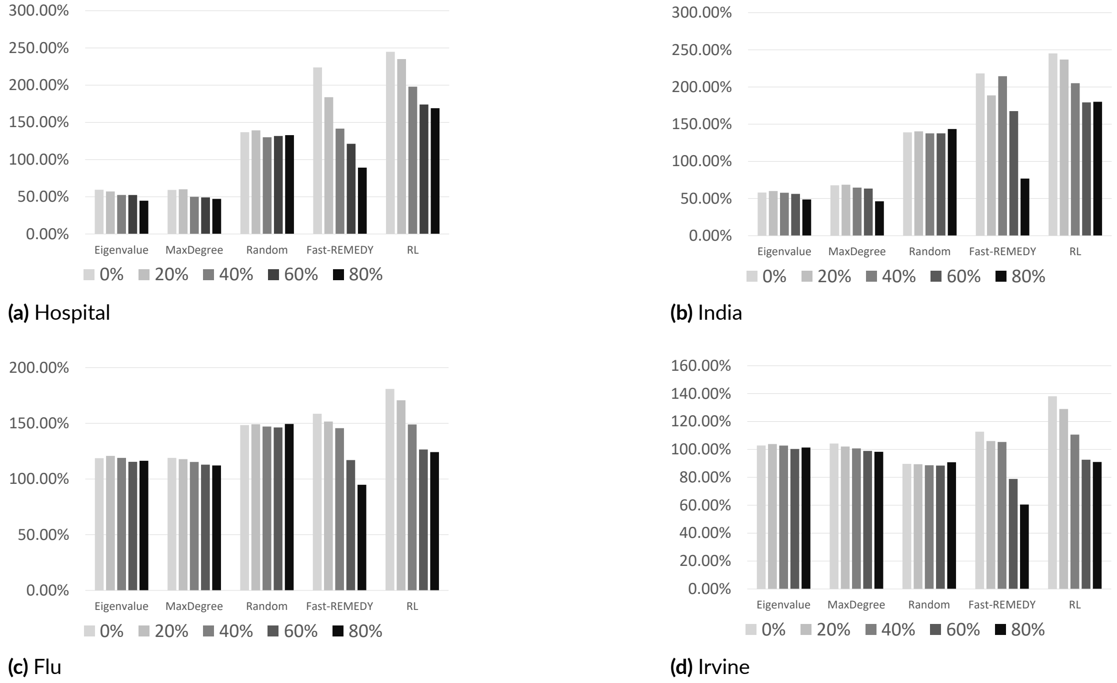
(a) Hospital

(b) India

(c) Flu

(d) Irvine

Figure 4.10: Performance under edge information removal.



(a) Hospital

(b) India

(c) Flu

(d) Irvine

Figure 4.11: Performance under node information removal.

5

Mobile Health Intervention

5.1 Problem Formulation

This chapter discusses the network mobile health intervention problem modeled as RMAB as another example of sequential network planning problems. However, unlike the active screening problem that only has a network effect for state transition, in this problem, we deal with interventions that have network effects. Even with fewer stochastic issues than the active screening problem, such a network effect makes the planning process of the mobile health intervention problem dramatically more complicated. I prove that the strong coupling between nodes makes the popular index policies not applicable and provides an alternative solution. We first introduce the general RMAB problems and their approaches.

5.1.1 General RMABs

. RMABs are a generalization of the well-studied multi-armed bandit model with many real-world applications. There are m arms $V = \{1, 2, \dots, m\}$; each arm $v \in V$ can be in one of several states $s_{v,t} \in \mathcal{S}$ at any time step $t \in \mathbb{N}$. At any time step, the decision maker can pull up to k arms. Each chosen arm v transitions in a Markovian fashion according to a transition matrix $\mathbf{P}^a$ and yields a reward $r_v(s_{v,t}) \geq 0$ that depends only on the state of the arm v at time t. In the restless setting, arms that are not chosen also transition, according to a different matrix $\mathbf{P}^p$. The elements $p^a_{s,s'}$ $(p^p_{s,s'})$ of the transition matrix capture the probability of transitioning from state s to s' when the arm is played (not played). Let $V_{a,t}$ denote the set of arms being played at time step t. The total reward of time step t can be expressed as $R_t = \sum_{v \in V_{a,t}} r_{v,t}(s_{v,t})$. Each arm can be described as a two-action Markov Decision Process (MDP) $(\mathcal{S}, \{0,1\}, \mathcal{R}, \mathcal{P})$. An action of 1 denotes that the arm is played and 0 that the arm is not played. Given the m MDPs and their initial states, the goal of this work is to find a policy for playing a sequence of k arms per round to maximize the average reward $\overline{R} = \lim_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T} R_t$.[*]

5.1.2 Networked RMABs for mobile interventions

. We consider a setting where each arm v corresponds to a location which has a population $n_v \in \mathbb{N}$. The state $s_v \in \mathcal{S} = \{0, \dots, n_v\}$ of a location is the number of healthy individuals. Individuals can either be in a healthy or, more generally, "good" state G or in a "bad" state B. Pulling an arm means visiting a location with a mobile intervention service, thereby exposing individuals at the location to

[*] Another frequently considered reward criterion is the discounted reward $\sum_{t=0}^{\infty} \beta^t R_t$ with $0 \leq \beta < 1$.

the intervention. We thus consider the transition matrices for individuals, depending on whether they receive an intervention ($\mathbf{P}_v^a$) or not ($\mathbf{P}_v^p$):

$$\mathbf{P}_v^a = \begin{array}{c c} & \begin{array}{cc} G & B \end{array} \\ \begin{array}{c} G \\ B \end{array} & \begin{bmatrix} 1 - p_{v,GB}^a & p_{v,GB}^a \\ p_{v,BG}^a & 1 - p_{v,BG}^a \end{bmatrix} \end{array}, \mathbf{P}_v^p = \begin{array}{c c} & \begin{array}{cc} G & B \end{array} \\ \begin{array}{c} G \\ B \end{array} & \begin{bmatrix} 1 - p_{v,GB}^p & p_{v,GB}^p \\ p_{v,BG}^p & 1 - p_{v,BG}^p \end{bmatrix} \end{array}. \qquad (5.1)$$

The transition probabilities are the same for all individuals with the same home location. Below, we will consider travel by individuals, which may result in them being exposed to the intervention at a different location. We stress that even in that case, an individual with home location v will transition according to the matrix P_v. This is because the characteristics of one's neighborhood are an important factor for one's health Ross & Mirowsky [2001], keeping in mind the intended application domains of the model. We assume that the transition probabilities and the initial states are known, but the transitions are not observed. This is because while population-level health data can be monitored, this rarely happens in real time. We omit subscripts when they are clear from the context.

In order to account for network effects from commuting (or more general travelling) behavior, we define a probability distribution for individuals over locations. Let $w_{u,v} \in [0,1]$ denote the probability that an individual with home location v is actually present in location u at any given moment (or that an individual from location v receives the intervention if location u is visited; we assume that individuals are sampled uniformly). Individuals can only be in one location at any given time, implying that $\sum_{u \in V} w_{u,v} = 1$. The matrix $\mathbf{W} \in [0,1]^{m \times m}$ with elements $w_{u,v}$ is the weighted adjacency matrix of the travelling network. Introducing the travelling network has two effects:

1. Not all individuals from location v are exposed to an intervention that visits v. In expectation, only $n_v w_{v,v}$ individuals from location v will receive the intervention (transition according to P_v^a)

due to a visit at location v. This property is an important extension of the recharging bandits model Kleinberg & Immorlica [2018]; in that model, it is assumed that each intervention fully "resets" the arm, i.e., puts all individuals into the good state.

2. Individuals from other locations receive the intervention when v is visited. In expectation, $\sum_{u \in V \setminus \{v\}} n_u w_{v,u}$ individuals from other locations receive the intervention at v.

The total number of individuals reached in any location thus depends on whether other locations are visited, and we define the vector $\mathbf{a}_t \in \{0,1\}^m$, with at most k elements equal to 1, to represent all actions taken in round t. The vector of expected fractions of the populations at each location v reached by an action vector $\mathbf{a}$ is given by $\hat{\mathbf{w}}(\mathbf{a}) = \mathbf{W} \cdot \mathbf{a}$. Letting $\hat{w}_v$ denote the v-th entry of $\hat{\mathbf{w}}$, we also define the weighted average transition probabilities for a location v as $\hat{\mathbf{P}}_v(\mathbf{a}) = \hat{w}_v(\mathbf{a}_t) \cdot \mathbf{P}^a_v + (1 - \hat{w}_v(\mathbf{a}_t)) \cdot \mathbf{P}^p_v$. Further let $\mathbf{s}_{v,t} = [s_{v,t}, n_v - s_{v,t}]$ be the total number of individuals in the good and bad state in location v at time t. By conditioning on the current state $\mathbf{s}_{v,t}$ and actions, we are able to obtain a closed form expression for the expected state in the next time step:

$$\begin{aligned}\mathbb{E}(\mathbf{s}_{v,t+1} \mid \mathbf{s}_{v,t}, \mathbf{a}_t, \ldots, \mathbf{a}_0) =& \mathbb{E}(\mathbf{s}_{v,t+1} \mid \mathbf{s}_{v,t}, \mathbf{a}_t) \\ =& \hat{w}_v \mathbf{s}_{v,t} \mathbf{P}^a_v + (1 - \hat{w}_v) \mathbf{s}_{v,t} \mathbf{P}^p_v \\ =& \mathbf{s}_{v,t} \hat{\mathbf{P}}_v(\mathbf{a}_t).\end{aligned}$$

However, the current state is unknown according to our assumptions. Hence we seek an expression for the expected future state that does not require knowledge of the current state. Consider the expected state at time t conditional only on the action history: $\mathbb{E}_t(\mathbf{s}_{v,t}) := \mathbb{E}(\mathbf{s}_{v,t} \mid \mathbf{a}_{t-1}, \ldots, \mathbf{a}_0)$. Using the law of total expectation, we obtain

$$\begin{aligned}\mathbb{E}_{t+1}(\mathbf{s}_{v,t+1}) =& \mathbb{E}(\mathbf{s}_{v,t+1} \mid \mathbf{a}_t, \dots, \mathbf{a}_0)\\ =& \mathbb{E}(\mathbb{E}(\mathbf{s}_{v,t+1} \mid \mathbf{s}_{v,t}, \mathbf{a}_t) \mid \mathbf{a}_t, \dots, \mathbf{a}_0)\\ =& \mathbb{E}(\mathbf{s}_{v,t}\hat{\mathbf{P}}_v(\mathbf{a}_t) \mid \mathbf{a}_t, \dots, \mathbf{a}_0)\\ =& \mathbb{E}(\mathbf{s}_{v,t} \mid \mathbf{a}_{t-1}, \dots, \mathbf{a}_0)\hat{\mathbf{P}}_v(\mathbf{a}_t),\end{aligned}$$

since $\mathbf{s}_{v,t}$ does not depend on $\mathbf{a}_t$ (only on previous actions). We thus obtain a recurrence relation for the expected state:

$$\mathbb{E}_{t+1}(\mathbf{s}_{v,t+1}) = \mathbb{E}_t(\mathbf{s}_{v,t})\hat{\mathbf{P}}_v(\mathbf{a}_t). \tag{5.2}$$

Eq. (5.2) allows us to compute the future expected state using only the current expectation and action vector. In order to fully describe the probability distribution of a single district, one would need $\binom{m}{k}$ matrices of size $(n_v + 1) \times (n_v + 1)$. Eq. (5.2) allows us to substantially reduce the complexity of the problem by focusing on the expected state. We write $\mathbb{E}_t(\mathbf{s}_{v,t}) = \mathbf{b}_{v,t}$ and use the recursion $\mathbf{b}_{v,t+1} = \mathbf{b}_{v,t}\hat{\mathbf{P}}_v(\mathbf{a}_t)$, where the initial state $\mathbf{b}_{v,0} = \mathbf{s}_{v,0}$ is known according to our assumptions.

The goal of the planner is to maximize the intervention benefit, taken as the sum of curing effects ($\text{cure}_v = p^a_{v,BG} - p^p_{v,BG}$) and prevention effects ($\text{prevention}_v = p^p_{v,GB} - p^a_{v,GB}$) for those individuals who received the intervention ($\text{cure}_v \hat{w}_v b_{v,t,2} + \text{prevention}_v \hat{w}_v b_{v,t,1}$, where $b_{v,t,1}$ and $b_{v,t,2}$ are the first and second element of $\mathbf{b}_{v,t}$, which are the expected total number of individuals in the good and bad state, respectively.), summed over locations and averaged over time steps. This criterion is chosen to align with the goals of applications such as MHCs which are to maximize the reach of a campaign Auerbach [2016], and to avoid underserving communities with a high probability of returning to the bad state, as could happen if only the total number of people in the good state ($R_t = \sum_{v\in V} s_{v,t}$)

were considered. Combining the curing and prevention effects, the reward per time step is given by: $R_t(\mathbf{a}_t) = \sum_{v \in V} \hat{w}_v(\mathbf{a}_t)\mathbf{s}_{v,t}(\mathbf{P}_v^a - \mathbf{P}_v^p) \cdot [1,0]^\top$. As discussed above, we focus on the expected reward and obtain:

$$\hat{R}_t := \mathbb{E}_t(R_t(\mathbf{a}_t)) = \sum_{v \in V} \hat{w}_v(\mathbf{a}_t)\mathbf{b}_{v,t}(\mathbf{P}_v^a - \mathbf{P}_v^p) \cdot [1,0]^\top. \tag{5.3}$$

We further make three assumptions that are natural in many relevant application domains; we combine assumptions made in prior work Mate et al. [2020] (assumptions (1) and (2)) with input from health experts (assumption (3)).

1. **The intervention is never bad for the individuals**: Health care interventions can help prevent disease or diagnose it early, reduce risk factors, and manage complications. Providing opportunities for increased access to quality services and interventions can reduce health disparities as well. Interventions provided via MHCs rarely result in negative impacts toward populations with little or no access to screening opportunities.

2. **The individuals are more likely to stay in the good state than to change from the bad state to good**: In most applications, moving to the good state (curing of a disease or access to food) is unlikely to happen spontaneously. Take oral health for example, Knowledge and implementation of proper preventive measures, such as brushing, flossing, getting dental cleanings, getting cavities filled timely, etc. is easier to do and less costly for both individual and the health care system, than treating or reversing diseases of the mouth once they have occurred.

3. **The curing effect of the intervention is larger then the prevention effect**: MHCs mostly serve otherwise under-served communities. Those who attend MHCs are typically concerned about their health and may already be exhibiting symptoms of underlying disease. Thus, identification of individuals with symptoms or with a disease may make interventions providing

services to reverse or manage disease more beneficial than interventions providing education on preventive measures more beneficial in these communities. This makes curing interventions generally more useful/desired than preventive measures. In food pantry applications, the prevention effect is typically small.

These assumptions are formalized in Eq. (5.4), for all $v \in V$:

$$p^p_{v,GB} \geq p^a_{v,GB} \text{ and } p^a_{v,BG} \geq p^p_{v,BG} \tag{5.4a}$$

$$1 - p^p_{v,GB} > p^p_{v,BG} \text{ and } 1 - p^a_{v,GB} > p^a_{v,BG} \tag{5.4b}$$

$$p^a_{v,BG} - p^p_{v,BG} > p^p_{v,GB} - p^a_{v,GB} \tag{5.4c}$$

Next, we show that these assumptions entail two properties that will prove useful later in constructing effective algorithms for the networked RMAB problem. Specifically, consider a district v, and suppose that there are no interventions in adjacent districts. We can then define the reward gain of visiting v after τ_v time steps as $H_v^{\text{upper}}(\tau_v, \hat{w}_v) = (p^p_{v,GB} - p^a_{v,GB})\hat{w}_v\hat{s}_{v,\tau_v} + (p^a_{v,BG} - p^p_{v,BG})\hat{w}_v(n^a_{v,\tau_v} - \hat{s}_{v,\tau_v})$ where $\hat{s}_{v,\tau_v}$ is the number of individuals in the good state at the time when the arm pull happens. This function has the following properties:

Theorem 3. *Under the assumptions in Eq.* (5.4)*, and assuming no interventions in neighboring districts, H^u_v is a monotone increasing concave function with respect to time τ_v elapsed since the last pull.*

Proof. We prove that the reward function of visiting a single location is a monotone increasing concave function with respect the the time elapsed since the arm was last pulled, assuming the initial state is better than the passive steady state ($\frac{s_0}{n} > \frac{p^p_{BG}}{p^p_{GB}+p^p_{BG}}$) and no neighboring arms are pulled. We proceed by proving: (1) Given $p^a_{v,GB} < p^p_{v,GB}$ and $p^a_{v,BG} > p^p_{v,BG}$ $\forall v \in V$ (intervention assumption 1), we always have $\frac{\hat{p}_{v,BG}}{\hat{p}_{v,GB}+\hat{p}_{v,BG}} > \frac{p^p_{v,BG}}{p^p_{v,GB}+p^p_{v,BG}}$ $\forall v \in V$. (2) Given $\frac{s_t}{n} \geq \frac{p^p_{BG}}{p^p_{GB}+p^p_{BG}}$ and $\frac{\hat{p}_{BG}}{\hat{p}_{GB}+\hat{p}_{BG}} > \frac{p^p_{BG}}{p^p_{GB}+p^p_{BG}}$, pulling the arm will always result in $\frac{s_{t+1}}{n} \geq \frac{p^p_{BG}}{p^p_{GB}+p^p_{BG}}$. (3) For any initial state $\frac{\hat{s}_0}{n} \geq \frac{p^p_{BG}}{p^p_{GB}+p^p_{BG}}$, $(1 - p^p_{GB} - p^p_{BG}) > 0$ and $\hat{p}_{BG}+\hat{p}_{GB}-p^p_{BG}-p^p_{GB} > 0$ (intervention assumption 2 and 3), the reward function is a monotone increasing concave function with respect to the time elapsed since the last pull.

We start by proving the first assertion: given $p^a_{v,GB} < p^p_{v,GB}$ and $p^a_{v,BG} > p^p_{v,BG}$, by our assumptions, we know that $\hat{p}_{v,GB} = \hat{w}_v p^a_{v,GB} + (1 - \hat{w}_v) p^p_{v,GB}$ for some $0 \leq \hat{w}_v \leq 1$ for any action taken. Thus we have $\hat{p}_{v,GB} < p^p_{v,GB}$ and $\hat{p}_{v,BG} > p^p_{v,BG}$. By rearranging the inequalities by

$$\hat{p}_{v,BG} \cdot p^p_{v,GB} + \hat{p}_{v,BG} \cdot p^p_{v,BG} > p^p_{v,BG} \cdot \hat{p}_{v,GB} + \hat{p}_{v,BG} \cdot p^p_{v,BG},$$

we obtain the conclusion (1):

$$\frac{\hat{p}_{v,BG}}{\hat{p}_{v,GB} + \hat{p}_{v,BG}} > \frac{p^p_{v,BG}}{p^p_{v,GB} + p^p_{v,BG}}.$$

Now, given $\frac{s_t}{n}$, the state after pulling the arm can be calculated as:

$$\frac{s_{t+1}}{n} = (1 - \hat{p}_{GB})\frac{s_t}{n} + \hat{p}_{BG}(1 - \frac{s_t}{n}).$$

If $\frac{s_t}{n} \leq \frac{\hat{p}_{BG}}{\hat{p}_{GB}+\hat{p}_{BG}}$, we have:

$$(\hat{p}_{GB} + \hat{p}_{BG} + 1 - 1)\frac{s_t}{n} \leq \hat{p}_{BG}$$

We can move some of the terms from the left to the right and obtain:

$$\frac{s_t}{n} \leq (1 - \hat{p}_{GB})\frac{s_t}{n} + \hat{p}_{BG}(1 - \frac{s_t}{n}) = \frac{s_{t+1}}{n}$$

Combining this with the condition $\frac{s_t}{n} \geq \frac{p^p_{BG}}{p^p_{GB}+p^p_{BG}}$, we get:

$$\frac{p^p_{BG}}{p^p_{GB}+p^p_{GB}} \leq \frac{s_t}{n} \leq \frac{s_{t+1}}{n}.$$

If $\frac{s_t}{n} > \frac{\hat{p}_{BG}}{\hat{p}_{GB}+\hat{p}_{BG}}$, using $(1-\hat{p}_{GB}-\hat{p}_{BG}) > 0$ (see intervention assumption 2) we have:

$$\begin{aligned}
\frac{s_{t+1}}{n} &= (1-\hat{p}_{GB})\frac{s_t}{n} + \hat{p}_{BG}(1-\frac{s_t}{n}) \\
&> (1-\hat{p}_{GB})\frac{\hat{p}_{BG}}{\hat{p}_{GB}+\hat{p}_{BG}} + \hat{p}_{BG}(1-\frac{\hat{p}_{BG}}{\hat{p}_{GB}+\hat{p}_{BG}}) \\
&= \hat{p}_{BG} + (1-\hat{p}_{BG}-\hat{p}_{GB})\frac{\hat{p}_{BG}}{\hat{p}_{GB}+\hat{p}_{BG}} \\
&= (\hat{p}_{BG}+\hat{p}_{GB})\frac{\hat{p}_{BG}}{\hat{p}_{GB}+\hat{p}_{BG}} + (1-\hat{p}_{BG}-\hat{p}_{GB})\frac{\hat{p}_{BG}}{\hat{p}_{GB}+\hat{p}_{BG}} \\
&= \frac{\hat{p}_{BG}}{\hat{p}_{GB}+\hat{p}_{BG}} \\
&> \frac{p^p_{BG}}{p^p_{GB}+p^p_{BG}},
\end{aligned}$$

which proves (2).

Finally, let $\frac{\hat{s}_\tau}{n}$ denote the fraction of individuals in the good state τ steps after an arm pull, and let $\frac{\hat{s}_0}{n}$ be its initial state. The reward function $H(\tau,\hat{w})$, where τ is the time since the last arm pull and $\hat{w}$ the share of the population exposed to an intervention, can be calculated as:

$$\begin{aligned}
H(\tau,\hat{w}) =& (p^p_{GB}-\hat{p}_{GB})n\hat{w}\frac{\hat{s}_\tau}{n} + (\hat{p}_{BG}-p^p_{BG})n\hat{w}(1-\frac{\hat{s}_\tau}{n}) \\
&= (\hat{p}_{BG}-p^p_{BG})\hat{w}n - (\hat{p}_{BG}+\hat{p}_{GB}-p^p_{BG}-p^p_{GB})\hat{w}n\frac{\hat{s}_\tau}{n}
\end{aligned}$$

The only variable here is $\frac{\hat{s}_\tau}{n}$ with a negative sign and positive coefficient $(\hat{p}_{BG}+\hat{p}_{GB}-p^p_{BG}-p^p_{GB})\hat{w}n$ (from intervention assumption 3). It is sufficient to prove that $\frac{\hat{s}_\tau}{n}$ is a monotone decreasing convex function. Given $\frac{\hat{s}_0}{n}$, using an eigendecomposition of the matrix $\mathbf{P}^p$, it can be written as:

$$\frac{\hat{s}_\tau}{n} = \frac{p^p_{BG}}{p^p_{GB} + p^p_{BG}} + (1 - p^p_{BG} - p^p_{GB})^\tau \cdot \left(\frac{\hat{s}_0}{n} - \frac{p^p_{BG}}{p^p_{GB} + p^p_{BG}} \right),$$

which is a monotone decreasing convex function given the intervention assumption 2 (which states that $(1 - p^p_{BG} - p^p_{GB}) > 0$). This proves the third assertion, and the theorem follows. □

Theorem 4. *Under the assumptions in Eq.* (5.4)*, and assuming no interventions in neighboring districts, H^u_v is a monotone increasing concave function with respect to the expected population share $\hat{w}_v$ exposed to the intervention.*

Proof. Consider two intervention schedules π_1 and π_2, whose respective intervention shares are given by $(\hat{w}_1(\pi_1), \hat{w}_2(\pi_1), \ldots, \hat{w}_T(\pi_1))$ and $(\hat{w}_1(\pi_2), \hat{w}_2(\pi_2), \ldots, \hat{w}_T(\pi_2))$. Given $\hat{w}_1(\pi_1) - \hat{w}_1(\pi_2) = \Delta w > 0$ and $\hat{w}_t(\pi_1) = \hat{w}_t(\pi_2)$ for all $t > 0$, we want to prove that π_1 always results in higher reward, assuming the same initial state of the node $\frac{s_0}{n}$. The total reward gain from one location can be written as

$$R(\pi) = \sum_{t=0}^{T} (p^p_{GB} - \hat{p}_{GB}(\pi)) s_t(\pi) + (\hat{p}_{BG}(\pi) - p^p_{BG})(n - s_t(\pi)).$$

The difference between the rewards of the two policies can thus be calculated as:

$$\begin{aligned} R(\pi_1) - R(\pi_2) = \Delta w &\left((p^a_{BG} - p^a_{GB}) - (p^p_{BG} - p^p_{GB}) \right) s_0 \\ &+ \sum_{t=1}^{T} \left((\hat{p}_{BG}(\pi) - \hat{p}_{GB}(\pi)) - (p^p_{BG} - p^p_{GB}) \right) \Delta s_t; \end{aligned}$$

here, Δs_t denotes the difference between the states induced by the two policies at time t. Let $\Delta \mathbf{b}_0 = [\Delta w s_0, -\Delta w s_0]^\top$, The Δs_t in each time step can be calculated as:

$$\Delta s_t = [1, 0] \prod_{\tau=0}^{t-1} \hat{\mathbf{P}} \Delta \mathbf{b}_0.$$

Observe that $\mathbf{b}_0$ happens to be an eigenvector of any $\hat{\mathbf{P}}$ with corresponding eigenvalue $(1 - \hat{p}_{GB}(\pi) - \hat{p}_{BG}(\pi))$. We have

$$\Delta s_t = \prod_{\tau=0}^{t-1} (1 - \hat{p}_{GB}(\pi) - \hat{p}_{BG}(\pi)) \Delta w s_0.$$

From intervention assumption 3, we can infer that $(1 - p^p_{GB} - p^p_{BG}) > (1 - \hat{p}_{GB}(\pi) - \hat{p}_{BG}(\pi))$. From intervention assumption 1, we can also infer that $p^a_{GB} - p^a_{BG} > \hat{p}_{GB}(\pi) - \hat{p}_{BG}(\pi)$ at any time step. Combining the above, we can infer that

$$\begin{aligned}
R(\pi_1) - R(\pi_2) &= \Delta w \left((p^a_{BG} - p^a_{GB}) - (p^p_{BG} - p^p_{GB}) \right) s_0 \\
&- \sum_{t=1}^{T} \left((\hat{p}_{BG}(\pi) + \hat{p}_{GB}(\pi)) - (p^p_{BG} + p^p_{GB}) \right) \Delta s_t \\
&> \Delta w s_0 \Big[(p^a_{BG} - p^a_{GB}) - (p^p_{BG} - p^p_{GB}) \\
&- \sum_{t=1}^{\infty} \left((\hat{p}_{BG}(\pi) + \hat{p}_{GB}(\pi)) - (p^p_{BG} + p^p_{GB}) \right) \cdot (1 - p^p_{GB} - p^p_{BG})^t \Big] \\
&> \Delta w s_0 \Big[(p^a_{BG} - p^a_{GB}) - (p^p_{BG} - p^p_{GB}) \\
&- \frac{(p^a_{BG} - p^a_{GB}) - (p^p_{BG} - p^p_{GB})}{p^p_{GB} + p^p_{GB}} \Big] \\
&= \Delta w s_0 \left((p^a_{BG} - p^a_{GB}) - (p^p_{BG} - p^p_{GB}) \right) \cdot \left(1 - \frac{1}{p^p_{GB} + p^p_{GB}} \right) \\
&> 0.
\end{aligned}$$

Thus, we have proved the theorem. □

Theorem 3 tells us that adding an extra pull to the intervention schedule of an arm will always improve the reward. From Theorem 4, we know that it is always preferable to intervene on a larger proportion of the population of an arm. These results suggest that the periodic policy is still a reasonable choice under the networked setting. The periodic policy in the non-networked setting is motivated by the following consideration: suppose that instead of pulling *exactly* k arms, we require only that *on average*, k arms are pulled in each round. In this relaxed problem, a periodic policy with suitable periods is optimal if the reward function is concave Kleinberg & Immorlica [2018].[†] Theorems 3 and 4 tell us that the reward function for the networked problem is still concave.

5.1.3 Why Modeling as Network RMAB Problem

Restlessness

This section shows that restlessness is essential for modeling the mobile clinic scheduling problem. One may argue that if the mobile health clinic could treat only a small portion of people in one location, its effect on the state of the location is negligible. Thus such a problem can be modeled using a simpler multi-arm bandit framework with no state transition. The optimal policy for such a simple model would be to find locations with the highest reward and repeatedly send vans there.

However, even if the fraction of people that mobile vans treat is small compared to the whole population, it may not be small compared to the number of people requiring mobile health van services. We show this by analyzing the data from Family Van and previous literature.

We looked into the data provided by Family Van and estimated the total population pool in need of the MHC service. We consider the log of the patient visiting survey collected from July 2019 to December 2020. There are $1,897$ different patients registered with different IDs across $8,835$ visit records. If we assume people in the population pool who demand MHC service visit the van uniformly at random, we can use the inverse coupon collector's problem formula to estimate the size of the total population that would come to the mobile clinic. Imagine that we have a bag with multiple unique balls with an unknown number, representing the total population that would come to the van we want to estimate. We draw from the bag $8,835$ times with replacement and find out there

[†] The constrained version is then a more difficult problem that involves solving a pinwheel problem, which is NP-hard.

are a total of $1,897$ types of unique balls in these $8,835$ samples. Based on the result from Dawkins [1991], we can calculate the total number of balls is most likely $1,951$. Note that this is the size of the population pool that demands service in the locations visited by the Family Van during a year (listed in table 5.1) instead of the entire area as in the previous example.

A similar result can be calculated from Stephanie et al. [2017] of children living with asthma in underserved populations. The sample size and unique IDs are $88,865$ and $15,986$ respectively on 4 mobile clinics in Southern California from November 1995 to December 2010. Again, using the formula of inverse coupon collector's problem, we estimate the size of the population pool demanding service is about $16,048$.

Although the data from previous examples are collected over a long period, it will not take long for the mobile health clinic to affect a large fraction of the target population. Based on a study in Northern Finland Pohjosenperä et al. [2019], only $6,622$ out of a population of $408,752$ are categorized in need of MHC service. These people live in multiple locations near the route of the 15 vans. From the data by Malone et al. [2020], mobile clinics provide a median number of $3,491$ visits annually. If we assume the total workday to be 240 days per year, we can estimate that each van serves about $3491/240 \approx 15$ people each day when they are in service. We again assume people visit the van uniformly at random and that each van serves about the same number of people. Thus each van has about $6622/15 \approx 441$ people needing its service. Each day, about 15 patients will visit the van uniformly at random. Based on the result of a variance of the coupon collector's formula $\sum_{i=0}^{T} \frac{441}{441-ki} \geq \frac{N}{2}$ with $(N,k) = (441, 15)$, we can solve for a minimum $T = 21$ days. This means that it would only take 21 days in expectation for these 15 vans to serve half of the target population by visiting their locations at least once, a large fraction of the target population that cannot be ignored.

All these examples point to the same conclusion: the fraction of people that mobile vans treat is not small compared to the potential service pool of each arm. Thus we can not ignore its action effect on the location state.

Recall that the optimal policy for the simpler model with no state transition is to repeatedly send vans to a fixed set of locations with the highest reward. However, due to the small service pool, it may not be a good strategy as even fewer populations can be reached by such a policy, especially when the service provided is one-shot or has low-frequency requirements like flu vaccination or cancer screening Group et al. [2002]. Furthermore, after enough time steps, if the van visits one location too frequently, most of the population would have already received treatment, and the low demand for service would reduce the number of patients who still need service, which may lead to a drop in the number of visiting patients. Thus the simpler model would lead to an optimal model which may be problematic to deploy. We avoid such issues by reflecting the effect of a recent visit by modeling the state of the arms in our model. In addition, we also want to capture the fact that the number of people visiting the clinic will eventually recover after enough timesteps. Thus we include the restlessness of passive transitions. As a result, our model captures the essential features and leads to a policy with the desired properties.

Network Effect

Previous literature and our analysis show that the commuting effect in mobile clinic service plays an essential role in serving the target populations. In a study done in rural Tanzania, Neke et al. [2018] show the average travel time of mobile clinic users is about 1.39 hours on average, with the maximum being 4.25 hours. According to the survey done in this study, the plurality of the population (about 41%) willing to travel more than 1.5 hours are those populations who feel they can not afford the cost of emergency care when they need it. In addition, in another study done in rural Mozambique Schwitters et al. [2015], the health service is so scarce that patients are willing to bike for a whole day, traveling up to 150km, and sleep overnight in a family member's house to get treatment the next day. In their study, some patients even worry that if more people from different locations learn about the van schedule, they will travel to the mobile health clinic and exhaust the service. These data indicate that

Table 5.1: Traveling distance samples of visiting patients of Family Van from July 2019 to December 2020

Locations of Vans	Average (miles)	Max (miles)	♯ of Logs	♯ of Logs $>$ 3 miles
Codman Square	2.060938	88.99839	225	38 (16.87%)
First Parish Church	18.56992	27.69679	20	19 (95.00%)
Mexican Consulate	34.71698	88.14121	46	40 (87%)
Nubian Square	4.560648	81.37295	171	95 (55.56%)
Salvadoran Consulate	3.858448	81.44579	292	101 (34.59%)
Upham's Corner	2.07965	44.94942	122	27 (22.13%)
Total	**5.242403**	**88.99839**	**876**	**320 (36.53%)**

there are indeed networks of commuting effects involved in mobile clinic service.

We have also analyzed the commuting distance using the Family Van survey data mentioned in the previous section. We use the self-report home zip code of the patients and the parking location of the van's service to estimate the commuting distance. After filtering out the data with missing or wrong zip codes, we have a total number of 876 effective sampling of the commuting distance. In table 5.1, we show the average commuting distance, maximum commuting distance, number of visiting logs, and number of visiting logs from more than 3 miles for each of the van's parking locations.

From the table, we observe that there is a large variation in the commuting range of different parking locations. For example, First Parish Church has most of its patients visiting within 30 miles, while Codman Square has the patient commuting from the furthest distance of almost 90 miles. Most of the far distances reported are from Nantucket island, where patients have to travel by boat to reach the van. Although most of the samples are close to the van's parking location, there are a total of 320 samples, which is about 37% of the samples that have their home locations further than 3 miles from the parking location. Thus, many people are willing to go from one location to another for more affordable medical services, leading to network effects in this problem.

5.2 Solution Approaches

As discussed previously, our problem shares significant similarities with the recharging bandits problem Kleinberg & Immorlica [2018]. Both in the network-free and networked setting, a natural solution approach is to (1) determine the frequencies with which arms should be pulled, and then (2) sequence the pulls optimally. Importantly, the network effects affect both stages of the solution approach. As a result, simple optimal (or near-optimal) policies from the non-networked setting may be far from optimal when networks are considered.

The fact that network effects must be taken into account in determining arm pull frequencies is easy to see. Consider a star graph in which the central node has population 0, while the $m-1$ leaf nodes have population $n_v = n$, and — importantly — have probability 1 of commuting to the central node. Without considering the network/commuting effect, any policy would choose a non-central node in each round (because the central node has population 0), whereas picking the central node in each round is clearly optimal.

Perhaps more interestingly, network effects also impact which sets of arms should be pulled simultaneously, even keeping the arm pull frequencies constant (and having identical arms). This is illustrated in the following example.

Example 1. *Consider the example shown in Fig. 5.1. We set $k = 2$ and $(p^p_{GB}, p^p_{BG}, p^a_{GB}, p^a_{BG}) = (p_{GB}, 0, p_{GB}, 1)$. All arms in Fig. 5.1 are identical. The optimal periodic policy is to select each arm every two rounds Kleinberg & Immorlica [2018]. Such a policy can be achieved without any rounding by selecting exactly two arms in each round. However, different ways of choosing these two arms result in policies with different rewards. Specifically, we consider the following two policies: Policy NN: Select two non-neighboring locations in each round. Policy NB: select two neighboring locations in each round. We also consider two different network scenarios with different commuting probabilities. In scenario 1, $w_{u,v} = \frac{1}{2}$ for all $(u,v) \in E$ and $w_{v,v} = 0$ for all $v \in V$, i.e., all individuals commute to adjacent nodes.*

In scenario 2, $w_{u,v} = \frac{1}{4}$ *for all* $(u, v) \in E$ *and* $w_{v,v} = \frac{1}{2}$ *for all* $v \in V$, *i.e., half of the individuals stay put. Table 5.2 summarizes the rewards of the two policies in the two scenarios: In scenario 1, the policy NN is the better policy for any* p_{GB}, *and the relative reward difference can be as large as* $\frac{2}{3}$. *In scenario 2, the policy NB becomes the better policy. For large* p_{GB}, *the relative reward difference approaches* $\frac{13}{15}$. *In particular, we see that the network effects must be taken into account in order to find the optimal way to coordinate the arm pulls of different arms.*

Table 5.2: Rewards of the two policies, and limits as $p_{GB} \to 1$, in the two scenarios.

	Scenario 1	Scenario 2
Policy NN	$\frac{4p_{GB}-2p_{GB}^2}{1+p_{GB}-p_{GB}^2} \to 2$	$\frac{4p_{GB}}{2p_{GB}+1} \to \frac{4}{3}$
Policy NB	$\frac{4p_{GB}}{2p_{GB}+1} \to \frac{4}{3}$	$\frac{52p_{GB}-32p_{GB}^2}{13+16p_{GB}-16p_{GB}^2} \to \frac{20}{13}$

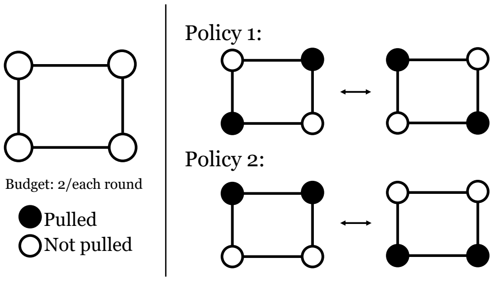


Figure 5.1: Example for how network combinatorial effects affect the reward of periodic policies.

Our proposed solution consists of two parts. In Section 5.2.1, we present an approach to obtain the optimal visiting period for each district. In Section 5.2.1, we illustrate our approach for synchronizing the arm pulls to optimize reward coupling.

Not only because of their schedule convenience, such approach could be beneficial for individuals who have already been screened by reinforcing the importance of treatment adherence and continued changes to their diet and lifestyle choices; changing behaviors related to diet and lifestyle require sustained efforts, long-term persistence, and, often, continued support and monitoring Willett et al. [2006]. Thus we consider a policy that cycles every T rounds.

5.2.1 Proposed approach

Despite the added model complexities compared to the non-networked Recharging Bandits model, our problem preserves similar concavity properties. In a similar vein as Kleinberg & Immorlica [2018], we thus aim to provide periodic policies for the networked RMAB problem, i.e., policies that repeat after T time steps. This not only facilitates scheduling, but can also reinforce intervention benefits in MHC domains Willett et al. [2006]. Exhaustively searching the action space of size $\binom{m}{k}^T$ is clearly impractical for reasonable problem sizes m. Fortunately, we can reduce the search space by exploiting the concavity we proved in Theorem 3.

Obtaining Visiting Periods

Let x_v be the fraction of times that arm v is chosen. When $1/x_v$ is integral, it can easily be shown that pulling the arm every $1/x_v$ rounds will maximize reward due to the concavity of the reward function Kleinberg & Immorlica [2018]. Define the period of pulling $\tau_v = 1/x_v \in \{1, 2, 3, \ldots, T\}$, meaning that v is visited every τ_v time steps. Let T be the maximum period considered, which could be a month, a season, or a year , depending on the application. Our goal is to find the optimal time period for each arm, subject to the sum of intervention frequencies being at most the budget $\sum_{v \in V} x_v \leq k$.

Suppose that a policy pulls arm v every τ_v time steps and follows some schedule $\pi : t \to \mathbf{a}_t$. We define $\mathbf{P}_v^*(\tau_v, \pi) = \prod_{t=0}^{\tau_v} \hat{\mathbf{P}}_v(\pi(t))$ as the transition matrix of the expected state vector right before the next arm pull. Note that the reward gained from pulling an arm v will depend on whether neighboring arms have recently been pulled, as this would imply that some share of v's population has already been exposed to the intervention. For a given τ_v, the reward gained from pulling v is minimized when all neighboring arms are visited in every round and maximized when no locations other than v are visited. We denote these two policies by π^ℓ and π^u, respectively. We can thus bound the average reward gained from pulling arm v every τ_v rounds (defined as $H_v(\tau_v)$) as:

$$\frac{1}{\tau_v}\overline{\mathbf{b}}_v^{\ell}\mathbf{P}_v^*(\tau_v, \pi^{\ell})\mathbf{n}_{v,G} \leq H_v(\tau_v) \leq \frac{1}{\tau_v}\overline{\mathbf{b}}_v^{u}\mathbf{P}_v^*(\tau_v, \pi^{u})\mathbf{n}_{v,G},$$

where $\overline{\mathbf{b}}_v^{\ell}$ ($\overline{\mathbf{b}}_v^{u}$) is the steady state of $\mathbf{P}_v^*(\tau_v, \pi^{\ell})$ ($\mathbf{P}_v^*(\tau_v, \pi^{u})$), which is also its eigenvector corresponding to its smallest eigenvalue. $\mathbf{P}_v^*(\tau_v, \pi)$ is the τ_v-step transition matrix of arm v given the policy of other arms π.

Given the upper bound $H_v^{\text{upper}}(\tau_v) = \frac{1}{\tau_v}\overline{\mathbf{b}}_{\mathbf{v}}^{\mathbf{u}}\mathbf{P}_v^*(\tau_v, \pi^{u})\mathbf{n}_{v,G}$, we can construct the reward table for each arm v by calculating the upper bound of each possible τ_v. Finding the optimal period for each arm thus becomes an optimization problem

$$\max \sum_{v \in V} H_v^{\text{upper}}(\tau_v) \qquad \text{s.t.} \sum_{v \in V} x_v \leq k.$$

We explicitly write the optimization problem as a MILP with integer variables $x_{v,t} \in \{0, 1\}$ for all $v \in V, t \in \{1, 2, \ldots, T\}$. $x_{v,t} = 1$ denotes that location v has a period of t. In the MILP, we write $H_v^u(t) := \frac{1}{t}\overline{\mathbf{b}}_{\mathbf{v}}^{\mathbf{u}}\mathbf{P}_v^*(t, \pi^{u})\mathbf{n}_{v,G}$ for all v and t.

$$\begin{array}{lll}
\text{Maximize} & R & \\
\text{subject to} & \sum_v \sum_{t=1}^{T} \frac{x_{v,t}}{t} \leq k & \text{(budget)} \\
 & \sum_{t=1}^{T} x_{v,t} \leq 1 \quad \text{for all } v & \text{(periods)} \\
 & R \leq \sum_{v \in V} \sum_{t=1}^{T} x_{v,t} H_v^u(t) & \text{(reward)} \\
 & x_{v,t} \in \{0, 1\} \quad \text{for all } v, t. &
\end{array} \tag{5.5}$$

The MILP (5.5) has $O(|V|T)$ constraints. Its implementation can be found in the source code provided. The first constraint captures that the chosen periods/frequencies allow a fractional solution of at most k visits per time step. The second set of constraints captures that each location has only one period. The third constraint bounds the reward. From the MILP solution, for each v, the period τ_v can be obtained as the (at most one) t such that $x_{v,t} = 1$. If $x_{v,t} = 0$ for all t for a particular v, then the arm is never worth pulling and can be discarded from the candidate pool.

The MILP can be adjusted to take fairness considerations into account as well. We list a few examples here; further details are discussed in the appendix:

- To achieve a minimum visiting frequency of $f_{\min}$, we can replace T with $T_{\min} = 1/f_{\min}$.
- To ensure that individuals from each node v have sufficient access to the intervention (either at v or a neighboring node), we can add the constraints $\sum_{u\in V}\sum_{t=0}^{T}\frac{w_{u,v}x_{u,t}}{t} \geq L$ for all v.
- To encourage the algorithm to increase the smallest node rewards, we can replace the reward with the alternative welfare function $R \leq \sum_{v\in V}\sum_{t=1}^{T} x_{v,t}(\frac{H_v^u(t)}{n_v})^\alpha/\alpha$ for $\alpha \leq 1$.

Finding optimal node sets to account for reward coupling

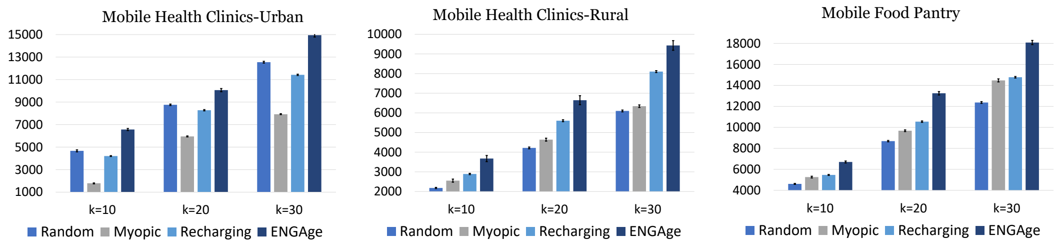
As illustrated in Example 1, the combinatorial effects of pulling arms in the networked RMAB problem induce reward coupling between the MDPs of the arms. In contrast to non-networked recharging bandits, the choice of which set of arms with equal optimal periods to pull in the same rounds thus matters in networked bandits. The potential loss in reward here stems from the fact that when two arms that are both neighboring arms of a third arm are intervened on in different time steps, they will deliver the intervention in part to the same individuals in the third arm.

In any time step t, for any pair of arms that is pulled simultaneously, we seek to maximize the overlap between the shares of populations in the set of arms that are neighbors of both arms. For a pair of arms (v, v'), this intervention overlap can be computed as $\sum_{u\in\partial(v)\cap\partial(v')} n_u w_{v,u} w_{v',u}$. If (and only if) the optimal periods τ_v and $\tau_{v'}$ are coprime to each other, this intervention overlap is independent of when the arms are intervened on. (As an example, two arms with periods 2 and 3 will be pulled together every six rounds, regardless of when the policy starts pulling each arm.) If the periods τ_u and

τ_v have a common factor, on the other hand, they can never be pulled together if they are out of sync. (Arms with periods 2 and 4 will never be pulled together if their sequences start one time step apart.) We would thus be losing out on the reward gains from pulling the arms together every $\mathrm{lcm}(\tau_u, \tau_v)$ rounds. In order to minimize this loss, we construct an undirected graph $\overline{G}(V, \overline{E})$ with the following edge weights:

$$\overline{w}_{v,v'}(\tau_v, \tau_{v'}) = \begin{cases} \sum_{u \in \partial(v) \cap \partial(v')} \frac{n_u w_{v,u} w_{v',u}}{\mathrm{lcm}(\tau_v, \tau_{v'})} & \text{if } \gcd(\tau_v, \tau_{v'}) > 1 \\ 0 & \text{otherwise} \end{cases} \tag{5.6}$$

The weight of the cut between the selected and unselected arms on $\overline{G}$ equals the average reward loss due to the intervention overlap. We can thus select the arm set to pull by minimizing the cut between the selected node set (of size k) and the unselected node set. Graph partition problems with node cardinality constraints are generally NP-hard Vazirani [2013]. We use a heuristic based on spectral graph partitioning, by considering the k nodes with the largest or smallest value in the eigenvector corresponding to the second-smallest eigenvalue of $\overline{L}$ (also known as the Fiedler vector), where $\overline{L}$ denotes the Laplacian of the graph $\overline{G}$. The ENGAge (Efficient Network Geography Aware scheduling) Algorithm (Algorithm 6) outputs an intervention policy based on this approach.



(a) MHCs in urban area **(b)** MHCs in rural area **(c)** Mobile Food Pantry

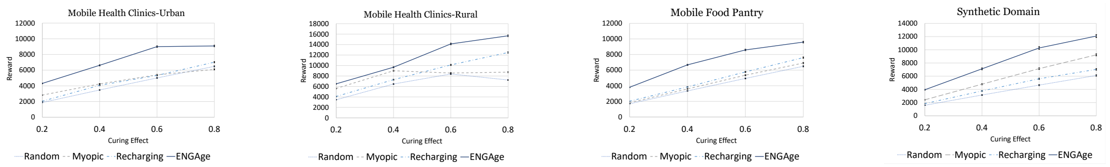
Figure 5.2: Average reward in three different domains under different budget constraints.

Algorithm 6 ENGAge

```
$V_{\text{candidate}} \leftarrow V$ and $V_{\text{wait}} \leftarrow \emptyset$.
Compute periods $\tau_v$ using the MILP (5.5).
Construct the new graph $\overline{G}(V, \overline{E})$ according to Eq. (5.6) and compute its Laplacian $\overline{L}$.
Find the set $\Lambda$ of Fiedler vectors of $\overline{L}$ (more than one in case of eigenvalue multiplicity).
for $t = 1, \ldots, T$ do
    for $v \in V_{\text{wait}}$ do
        $\text{Timer}(v) \leftarrow \text{Timer}(v) - 1$.
        if $\text{Timer}(v) = 0$ then
            Move $v$ from $V_{\text{wait}}$ to $V_{\text{candidate}}$.
        end if
    end for
    $V_a(t) \leftarrow \emptyset$.
    for all $\eta \in \Lambda$ do
        Find the sets of nodes with $k$-th largest and smallest elements in $\eta$: Specifically, let
        $\eta_{(k)}$ denote the $k$-th largest entry of $\eta$, set $\overline{V} \leftarrow \{v \in V_{\text{candidate}} \mid \eta_v \leq \eta_{(k)}\}$ and
        $\underline{V} = \{v \in V_{\text{candidate}} \mid \eta_v \geq \eta_{(m-k+1)}\}$.
        If $|\overline{V}| > k$ or $|\underline{V}| > k$, reduce the set size to $k$ by arbitrarily removing tied nodes at
        the cutoff threshold.
        Update $V_a(t)$ to the set $S$ that minimizes the cut: $V_a(t) \leftarrow$
        $\text{argmin}_{S \in \{\overline{V}, \underline{V}, V_a(t)\}} c(S)$. Here, $c(S)$ denotes the cut capacity of the node set
        $S$ in $\overline{G}$ (and is defined as $\infty$ for the empty set). Arbitrarily break ties.
    end for
    Move $V_a(t)$ from $V_{\text{candidate}}$ to $V_{\text{wait}}$, and set $\text{Timer}(v) \leftarrow \tau_v$ for these arms.
end for
return $V_a(t)$ as arms to pull at time $t$ for all times $t = 1, \ldots, T$.
```

5.2.2 Analysis

We start by analyzing the complexity of the solution approach described above. The concave MILP (5.5) can be solved efficiently using time $O(|V|T\log(|V|T))$, by sorting the set of slopes of segments, corresponding to the different $H_v^u(t)$. Details are given in Kleinberg & Immorlica [2018]. In our implementation, we instead use an off-the-shelf MILP solver. While its worst-case running time is larger, as our experiments show, it runs very efficiently in practice. Calculating the Laplacian $\overline{L}$ requires find-



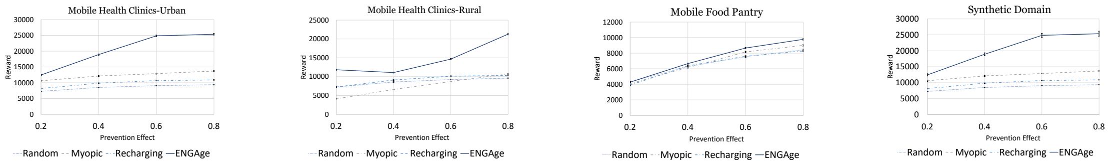
(a) Reward for different curing rates in urban MHC domain

(b) Reward for different curing rates in rural MHC domain

(c) Reward for different curing rates in food pantry domain

(d) Reward for different curing rates in synthetic domain

Figure 5.3: Average reward in three different domains under different curing effects ($P^a_{GB} - P^p_{GB}$) .



(a) Reward for prevention rates in urban MHC domain

(b) Reward for prevention rates in rural MHC domain

(c) Reward for prevention rates in food pantry domain

(d) Reward for prevention rates in synthetic domain

Figure 5.4: Average reward in three different domains under different prevention effects ($P^p_{BG} - P^a_{BG}$) .

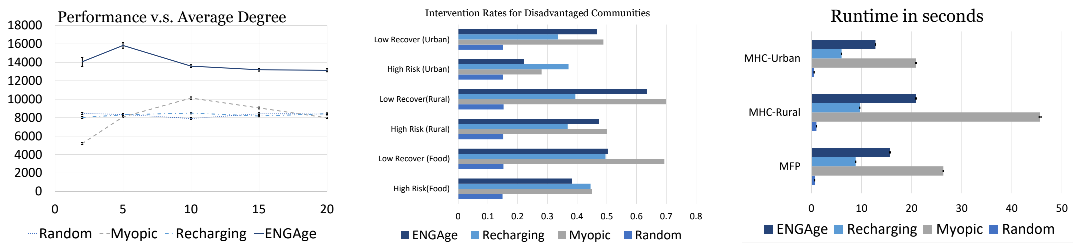
ing common neighbours ($O(\hat{d}|E|)$ by An et al. [2019], where $\hat{d}$ is the maximum degree in $\overline{G}$) and then computing their gcd ($O(\log T)$ using the Euclidean algorithm). The overall cost of computing the Laplacian is thus $O(\hat{d}|E| \log T)$. Again, our actual implementation is less efficient in terms of worst-case complexity, but runs fast in practice nonetheless. Finding a Fiedler vector takes time $O(\overline{d}|V|)$ using Lanczos' algorithm Lanczos [1950], where $\overline{d}$ is the average degree of $\overline{G}$. The rest of the planning takes time $O(|V|T)$. Thus, the total time complexity of our algorithm is $O(|V|T\log(|V|T) + \hat{d}|E| \log T)$ for the dominant term. Other than being efficient to compute, the schedule our algorithm outputs is likely to have little variance on visiting district v every τ_v rounds. The underserved individuals in the area can thus anticipate the intervention visit easier and benefit from the reinforcing effect mentioned earlier in this section.

In Section 5.3, we experimentally evaluate the performance of our algorithm on various graphs from real-world domains. We now turn to analyzing sufficient conditions that guarantee optimality for various cases that we will discuss below.

First consider the case of homogeneous nodes and edge weights, i.e., all nodes have the same populations and transition probabilities between states, and all edges have the same commute probabilities. If we replace the eigenvector-based heuristic in ENGAge with an oracle that optimally solves the min-cut problem with cardinality constraints, then ENGAge outputs the optimal policy for arbitrary graphs of N nodes whenever $k|N$. This is because in this case, the cut on the constructed graph measures the exact reward loss of the schedule. Solving the min-cut problem optimally will then lead to the optimal scheduling.

Next, consider the special case in which the graph $\overline{G}$ has γ connected components $C_1, \ldots, C_\gamma$, each of size $|C_i| = k$. Furthermore, we assume that all elements of the same component have the same optimal period; that is, if $u, v \in C_i$, then $\tau_u = \tau_v$. For $\gamma \geq 2$, note that $\overline{L}$ is positive semidefinite as $\overline{G}$ is undirected for arbitrary input graphs G by construction. The smallest eigenvalue 0 will have multiplicity γ in the Laplacian $\overline{L}$. Thus, $|\Lambda| = \gamma$, and it is known that each component C_i has a corresponding Fiedler vector supported entirely on C_i Marsden [2013]. Hence, in each iteration, Algorithm 6 will select exactly all members of one component. As there are no links between nodes in different components by definition, all members of a component will be fully intervened on. Our problem thus reduces to a pinwheel problem with γ arms and optimal periods τ_i for $i = 1, \ldots, \gamma$. Pinwheel problems are known to be NP-hard in general Chan & Chin [1993], but optimal solutions are known to exist in special cases where all periods are multiples of one another and $\sum_i^\gamma \tau_i \leq k$ Holte et al. [1989]. The optimal solution in these cases can be obtained by a simple greedy policy (see Chan & Chin [1993]) which is realized by the sets V_{wait} of our algorithm. The latter condition is guaranteed by the setup of ENGAge; hence, our proposed approach will output an optimal schedule in those cases. For $\gamma = 1$, the same conclusion follows trivially, because the algorithm can visit all locations in each time step.

Based on the above analysis, ENGAge will output the optimal policy in the following settings, among others: (1) Complete graphs with equal edge weights, identical nodes, and $k|N$. (2) Graphs with multiple connected components, each of size k, with equal edge weights and identical nodes. (3) Rings with edge weights $1/2$, identical nodes, and $k = N/2$. (4) d-dimensional Hypercubes with edge weights $1/d$, identical nodes, and $k = N/d$. (5) Bipartite or multipartite graphs with partitions of size k, identical node degrees, and edge weights summing to 1 for all nodes. (6) Strongly d-regular graphs with equal edge weights and identical nodes. These are illustrative examples of graphs where our algorithm is guaranteed to perform optimally. In the next section, we will empirically show that it outperforms existing methods in more general settings, including real-world graphs.



(a) Average Degree

(b) Disadvantaged Communities

(c) Runtime

Figure 5.5: (5.5a): average reward vs. graph average degree (5.5b): intervention rates for 15% most disadvantaged communities. (5.5c): average runtimes.

5.3 Experimental Evaluation

We perform experiments comparing our algorithm to baselines in a variety of real-world application scenarios. We begin by describing the application domains and their properties:

Mobile Health Clinics in urban areas: This domain setting is modeled on MHCs that are an important part of urban health care programs. Specifically, we consider a graph of the city of Boston (where such MHCs are used by non-profit organizations Chen et al. [2022]), collected from Boeing [2017]. The graph consists of 431 locations that are used as bandit arms. The populations n_v and

Table 5.3: Properties of the network data sets.

Network	$\lvert V\rvert$	average degree	average degree centrality
Boston	431	2.92	0.005
Daniels County	631	2.53	0.008
Los Angeles	561	2.85	0.001

transition probabilities $(p^p_{vGB}, p^p_{vBG}, p^a_{vBG}, p^a_{vGB})$ are generated from uniformly random distributions subject to the assumptions introduced in the problem formulation section[‡].

Mobile Health Clinics in rural areas: In contrast to urban areas, rural areas are characterized by a larger number of less connected smaller communities, and may experience lower overall levels of access to health services. We model this domain using a graph of Daniels County, MT, with 631 locations, taken from Boeing [2017]. Daniels County is considered one of the most rural counties in the US, as measured by the index of relative rurality Waldorf [2007]. We modify the previous setting to set a large portion of districts to have communities with relatively small population, to account for the characteristics described before.

Mobile Food Pantry: Due to a limited choice of means of transportation, residents of many socially disadvantaged neighborhoods can only access food within shorter distances; as a result, healthy food options are often limited. Mobile food pantries (MFPs) have become an important source of healthy food for these communities Algert et al. [2006]. In the MFP scenario, the Los Angeles city graph with 561 locations collected from Boeing [2017] is used, as food insecurity is an important issue in Los Angeles. In this scenario, it is assumed that there is no prevention effect ($p^a_{GB} = p^p_{GB}$), as the provided food needs to be fresh and will only be distributed to individuals in bad states.

We compare our algorithm to three baseline algorithms. Random selects k locations uniformly at random in each time step. Myopic selects the locations with maximum reward in the current time

[‡] While we have access to real-world street graph data, we do not have access to population and commuting data at a matching granularity.

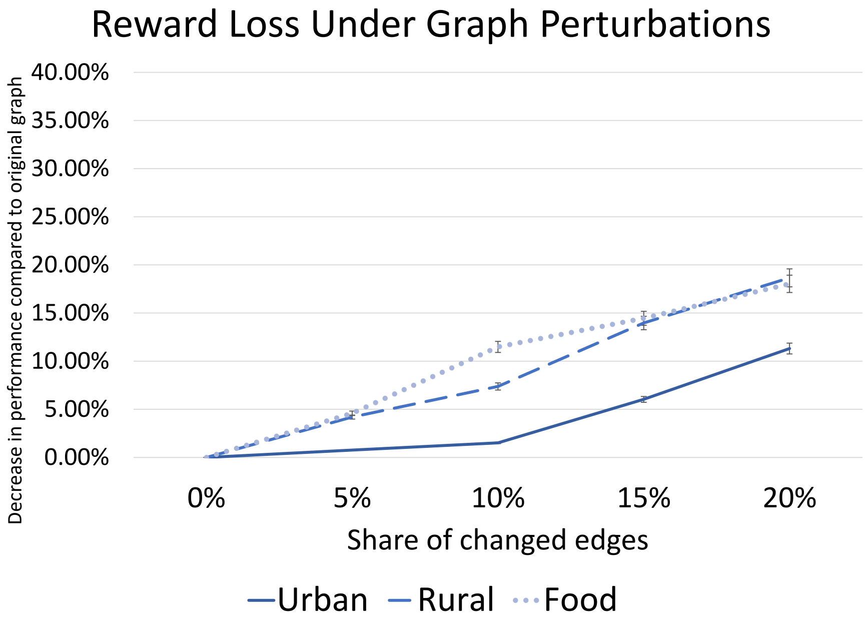


Figure 5.6: Sensitivity analysis.

step. Recharging is the rounding scheme scheduling provided in Kleinberg & Immorlica [2018]. All experiments are conducted on a system with 6 cores, 2.60 GHz Intel CPU, and 16 GBs of RAM for 30 simulations over 100 time steps for each trial. All figures include approximate 95% confidence intervals as error bars. Figures 5.2a–5.2c show the average reward collected with different budgets of $k \in \{10, 20, 30\}$ arms, for the three domains described above. Our algorithm consistently outperforms all baselines. Recharging mostly performs second-best, though in the urban MHC setting, it is slightly worse than Random. Figure 5.5c shows the average runtime per simulation in seconds. Interestingly, Myopic is the slowest algorithm, because it has to compute the reward for each node in each round, while ENGAge and Recharging use pre-computed period tables.

We further analyze the sensitivity of these results to several modeling parameters. Figure 5.5a shows the performance of the algorithms for different densities for a synthetic domain based on a spatial preferential attachment model Barthélemy [2011], Ferretti & Cortelezzi [2011]. The results are nonmonotonic for the ENGAge algorithm. A possible explanation could be that there might exist a level of optimum connectivity, below which adding more links will increase the intervention benefit by spreading interventions more widely, and above which adding more links will cause too much overlap between the populations that are intervened on in different time steps. Figures 5.3 and 5.4 show that ENGAge consistently outperforms the baselines across multiple values for cure and prevention rates in all domains.

We also analyze the impact of our algorithm on the most disadvantaged communities, i.e., those experiencing the highest risk of transitioning to the bad state, or which have small probability of recovering from the bad state. Figure 5.5b shows the average intervention frequencies for the 15% communities with the highest risk (p^p_{GB}) and lowest chance of recovery (p^p_{BG}). All algorithms except Random intervene on the most disadvantaged communities disproportionately more often, showing that they are not discriminating against them. This is thanks to the design of the reward criterion that measures intervention benefit for individuals receiving the intervention.

Finally, we conduct a sensitivity analysis of the ENGAge algorithm against graph perturbations. Figure 5.6 is constructed as follows: Starting with the real-world graphs from the three domains, we add perturbations by removing a given percentage of the edges, and adding back the same number of edges randomly. In the optimization, we then use the perturbed graph, while the original, unperturbed graph is used to compute the rewards. Overall, we observe that perturbing x% of edges generally reduces reward by less than x%. For example, with a graph perturbation of 15%, the performance reductions in the urban, rural and food settings are 6%, 13%, and 14%, respectively.

5.4 Summary

This chapter present a networked RMAB model motivated by mobile interventions; our model captures network effects stemming from traveling behavior. Our model was built based on the input of domain experts in mobile health interventions. To the best of our knowledge, this is the first paper addressing the challenge of scheduling multiple interventions with network effects in the RMAB model. Network effects induce strong reward coupling between arms, substantially complicating the analysis of the RMAB. We propose the ENGAge (Efficient Network Geography Aware scheduling) algorithm that takes reward coupling and network effects into account. We provide sufficient conditions for optimality and show that our algorithm outperforms several baselines empirically in three real-world domains and synthetic domains with varying properties.

6

Conclusion

6.1 Contributions

The rapid advance of artificial intelligence has made new applications possible with the help of domain expertise. In particular, public health has gained more and more attention due to growing health consciousness and pandemic outbreaks in recent years. Previous research has considered numerous approaches to applying AI to improve the practice of different public health applications. However, due to the technical limitation, they all focus on more specific and ideal settings with either one-shot optimization, no network structure, or fully observable states. My thesis explores and expands AI techniques on different sequential network planning problems with various challenges. Such planning problems have a wide range of application scenarios in the public health domain, yet AI literature has not explored much due to the previous technique limitations. This thesis formalizes the active screening and mobile health intervention model as two different sequential network planning problems. Moreover, my thesis takes the forward step of solving these planning problems by overcoming the challenges that the previous state of the arts could not handle.

First, my thesis introduces the active screening problem. This problem involves solving a sequential planning problem with network transitions under state uncertainty. Active screening offers a powerful yet expensive means to control disease spread in the public health domain that passive screening cannot achieve due to its latency of cure. Thus, this model provides a basis for developing a screening strategy to minimize the spread of recurrent diseases. Furthermore, unlike previous literature that develop one-shot network optimization problems for non-recurrent diseases, this model is applicable when one cannot permanently cure the disease or the vaccination is not immediately available. The proposed model also considers real-world constraints such as uncertain health states and limited intervention resources. Unfortunately, this complicated problem turns out to be NP-hard shown by my proof. To solve this problem, I propose two novel algorithms, Full- and Fast-REMEDY. Full-REMEDY considers the effect of future actions and provides high solution quality, whereas Fast-REMEDY scales linearly in the size of the network. I examined the effectiveness of the REMEDY algorithm on several real-world datasets which emulate human contact against different baselines. It shows superior performance on both speed and degree of infectious number reduction in all scenarios. In addition, I also show that REMEDY is robust to errors in estimates of disease parameters and incomplete information about the contact network.

Second, while REMEDY is effective for slow-spreading diseases like Tuberculosis, its full version is not scalable for fast-path diseases that require planning for an immense horizon, while the fast version does not consider the future effect of the current action. I propose a novel reinforcement learning (RL) approach based on Deep Q-Networks (DQN) to get the best for both worlds. Applying RL to the active screening is not a simple task for several reasons. Besides the need for sequential planning and the uncertainties in the infectiousness states of the population, the combinatorial nature of the action choice and the sparseness of the reward makes the standard RL approaches hard to converge. To overcome these challenges, first, I use graph convolutional networks (GCNs) to represent the Q-function that exploits the node correlations of the underlying contact network. Second, to avoid solving a combinatorial optimization problem in each period, I decompose the node-set selection as a sub-sequence of decisions and further design a two-level RL framework that solves the problem in a hierarchical way. Finally, to speed up the slow convergence of RL, which arises from reward sparseness, I incorporate ideas from curriculum learning into my hierarchical RL approach. In evaluating my RL algorithm on several real-world networks, results show that my RL algorithm can scale up to 10 times the problem size of Full-REMEDY in terms of planning time horizon. Meanwhile, it outperforms Fast-REMEDY by up to 33% in solution quality.

Lastly, my thesis introduces the mobile health intervention problem, another example of public health application for sequential network planning problems. Motivated by a broad class of mobile intervention problems, I propose and study restless multi-armed bandits (RMABs) with network effects. Unlike the active screening problem and most previous network problems, the network effect in this sequential planning problem is on the intervention instead of just state transition. In this model, arms are partially recharging and connected through a graph so that pulling one arm also improves the state of neighboring arms, significantly extending the previously studied setting of fully recharging bandits with no network effects. In mobile interventions, network effects may arise due to regular population movements (such as commuting between home and work). I show that network effects in RMABs induce strong reward coupling that is not accounted for by existing solution methods. I propose a new solution approach for networked RMABs, exploiting concavity properties that arise under natural assumptions on the structure of intervention effects. We provide sufficient conditions for optimality of our method in idealized settings and demonstrate that it empirically outperforms state-of-the-art baselines in three mobile intervention domains using real-world graphs.

6.2 Future Directions

My thesis has opened up a large avenue for applying sequential network panning problems to various public health scenarios. In this thesis, I have explored two critical applications with different challenges. However, other challenges may emerge as we extend the model I propose to align with the real world. For instance, while this work has addressed the uncertainty of the state transition, there may be other uncertainties involved in the network sequential planning process pipeline. There might be network structure uncertainty due to missing data. There might be intervention uncertainty due to the absence of participants or the effectiveness of the interventions. Accounting for these uncertainties may further improve the effectiveness of the decision-making. Furthermore, how to combine the information we have and adjust our policy as our observations unfold while these uncertainties present are yet to be considered by the literature.

Additionally, exploiting the network structure will be an essential direction for future work. In this thesis, I have made very few assumptions about network structure. While this makes the solution proposed more general, it could be the case that there is room for improvement by making reasonable assumptions about the network structure. There is also the potential to adapt the solution I proposed to a larger class of problems by transforming them into a graph, like the matching problem into a bipartite graph. Although different challenges might emerge, the insight about future planning and combinatorial effect uncovered in this work will undoubtedly be helpful when developing the solutions.

In terms of future work regarding active screening for tuberculosis, one obstacle between implementation and real-world deployment is the costly screening process. Screening for tuberculosis involves getting people to X-ray machines, which can be time-consuming due to a lack of resources and limited diagnosis speed. However, other than contact networks, we can use cell phones to further collect cough sounds and other meta-data. Wadhwani AI, the organization in India I worked closely with for the active screening project, is developing a cough-based screening tool for tuberculosis. The goal is to make it usable with a simple cell phone. Combining contact network, cough sound, and other meta-data could potentially realize rapid screening and represent an excellent test-based for the active screening algorithm in the future.

As for mobile health scheduling, more efforts are still needed to expand service availability to low resource communities further. In this thesis, distances from the potential patients to the mobile health clinic have been modeled to be an important factor based on the input of health experts. According to their data, most of the visited patients are within 25 minutes of communication time through walking or driving. However, other factors affect the availability of MHC services to specific populations. For example, one observation made by our partner health experts from the data is that women are $30 \sim 35\%$ more likely to use the MHC services than men. We hypothesize that men in low resource communities usually need to communicate far distances for better work opportunities. Therefore, it is unlikely to visit the MHC during regular working days. Addressing issues like this by accounting for different visiting times or identifying more potential patients are all interesting areas for future work of real-world deployment beyond my thesis.

References

Abbou, A. & Makis, V. (2019). Group maintenance: A restless bandits approach. *INFORMS Journal on Computing*, 31(4), 719–731.

Adelman, D. & Mersereau, A. J. (2008). Relaxations of weakly coupled stochastic dynamic programs. *Operations Research*, 56(3), 712–727.

Akbarzadeh, N. & Mahajan, A. (2019). Restless bandits with controlled restarts: Indexability and computation of whittle index. In *2019 IEEE 58th Conference on Decision and Control (CDC)* (pp. 7294–7300).: IEEE.

Algert, S. J., Agrawal, A., & Lewis, D. S. (2006). Disparities in access to fresh produce in low-income neighborhoods in los angeles. *American journal of preventive medicine*, 30(5), 365–370.

An, X., Gabert, K., Fox, J., Green, O., & Bader, D. A. (2019). Skip the intersection: Quickly counting common neighbors on shared-memory systems. In *2019 IEEE HPEC* (pp. 1–7).: IEEE.

Anderson, R. M. & May, R. M. (1992). *Infectious diseases of humans: dynamics and control*. Oxford University: Oxford University.

Arulkumaran, K., Deisenroth, M. P., Brundage, M., & Bharath, A. A. (2017). Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine*, 34(6), 26–38.

Auerbach, J. (2016). The 3 buckets of prevention. *Journal of public health management and practice: JPHMP*, 22(3), 215.

Ayer, T., Zhang, C., Bonifonte, A., Spaulding, A. C., & Chhatwal, J. (2019). Prioritizing hepatitis c treatment in us prisons. *Operations Research*, 67(3), 853–873.

Bailey, N. T. (1975). *The mathematical theory of infectious diseases and its applications*. Charles Griffin & Company Ltd: Charles Griffin & Company Ltd.

Ball, F. G., Knock, E. S., & O'Neill, P. D. (2015). Stochastic epidemic models featuring contact tracing with delays. *Mathematical biosciences*, 266, 23–35.

Banerjee, A., Chandrasekhar, A. G., Duflo, E., & Jackson, M. O. (2013). The diffusion of microfinance. *Science*, 341(6144), 1236498.

Bansal, S., Grenfell, B. T., & Meyers, L. A. (2007). When individual behaviour matters: homogeneous and network models in epidemiology. *Journal of the Royal Society Interface*, 4(16), 879–891.

Barthélemy, M. (2011). Spatial networks. *Physics Reports*, 499(1-3), 1–101.

Bengio, Y., Louradour, J., Collobert, R., & Weston, J. (2009). Curriculum learning. In *ICML* (pp. 41–48).

Bernoulli, D. & Blower, S. (2004). An attempt at a new analysis of the mortality caused by smallpox and of the advantages of inoculation to prevent it. *Reviews in medical virology*, 14(5), 275–288.

Bertsimas, D. & Niño-Mora, J. (2000). Restless bandits, linear programming relaxations, and a primal-dual index heuristic. *Operations Research*, 48(1), 80–90.

Biswas, A., Aggarwal, G., Varakantham, P., & Tambe, M. (2021). Learn to intervene: An adaptive learning policy for restless bandits in application to preventive healthcare. *2021 IJCAI*.

Boeing, G. (2017). U.s. street network. In *U.S. Street Network Shapefiles, Node Edge Lists, and GraphML Files*. Harvard Dataverse.

Braxton, J., Davis, D. W., Emerson, B., Flagg, E. W., Grey, J., Grier, L., Harvey, A., Kidd, S., Kim, J., Kreisel, K., et al. (2017). Sexually transmitted disease surveillance. *CDC*.

Cadman, D., Chambers, L., Feldman, W., & Sackett, D. (1984). Assessing the effectiveness of community screening programs. *Jama*, 251(12), 1580–1585.

CDC (2011). Tuberculosis: General information. *MMWR. Recommendations and reports: Morbidity and mortality weekly report*.

Chan, M. Y. & Chin, F. (1993). Schedulers for larger classes of pinwheel instances. *Algorithmica*, 9(5), 425–462.

Chen, H., Ghosh, S., Fan, G., Behari, N., Biswas, A., Williams, M., Oriol, N. E., & Tambe, M. (2022). Using public data to predict demand for mobile health clinics. In *In 2022 IAAI*.

Chen, W., Wang, Y., & Yang, S. (2009). Efficient influence maximization in social networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 199–208).: ACM.

Chinnakali, P., Thekkur, P., Ramaswamy, G., Selvaraj, K., et al. (2016). Active screening for tuberculosis among slum dwellers in selected urban slums of Puducherry, South India. *Annals of Tropical Medicine and Public Health*, 9(4), 295.

Danon, L., Ford, A. P., House, T., Jewell, C. P., Keeling, M. J., Roberts, G. O., Ross, J. V., & Vernon, M. C. (2011). Networks and the epidemiology of infectious disease. *Interdisciplinary perspectives on infectious diseases*, 2011.

Dawkins, B. (1991). Siobhan's problem: the coupon collector revisited. *The American Statistician*, 45(1), 76–82.

Dayan, P. & Hinton, G. E. (1993). Feudal reinforcement learning. In *NeurIPS* (pp. 271–278).

Deo, S., Iravani, S., Jiang, T., Smilowitz, K., & Samuelson, S. (2013). Improving health outcomes through better capacity allocation in a community-based chronic care model. *Operations Research*, 61(6), 1277–1294.

Dickerson, J. P., Procaccia, A. D., & Sandholm, T. (2012). Optimizing kidney exchange with transplant chains: Theory and reality. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 2* (pp. 711–718).

Dietterich, T. G. (2000). Hierarchical reinforcement learning with the maxq value function decomposition. *Journal of AI research*, 13, 227–303.

Dong, W., Heller, K., & Pentland, A. S. (2012). Modeling infection with multi-agent dynamics. In *International Conference on Social Computing, Behavioral-Cultural Modeling, and Prediction* (pp. 172–179).: Springer.

Drakopoulos, K., Ozdaglar, A., & Tsitsiklis, J. N. (2014). An efficient curing policy for epidemics on graphs. *IEEE Transactions on Network Science and Engineering*, 1(2), 67–75.

Drakopoulos, K., Ozdaglar, A., & Tsitsiklis, J. N. (2016). When is a network epidemic hard to eliminate? *Mathematics of Operations Research*.

DuBois, T., Eubank, S., & Srinivasan, A. (2012). The effect of random edge removal on network degree sequence. *Electronic journal of combinatorics*, 19(1).

Eames, K. T. & Keeling, M. J. (2003). Contact tracing and disease control. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 270(1533), 2565–2571.

Ferretti, L. & Cortelezzi, M. (2011). Preferential attachment in growing spatial networks. *Physical Review E*, 84(1), 016103.

Frank, M. & Wolfe, P. (1956). An algorithm for quadratic programming. *Naval research logistics quarterly*, 3(1-2), 95–110.

Ganesh, A., Massoulié, L., & Towsley, D. (2005). 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 (pp. 1455–1466).: IEEE.

Garetto, M., Gong, W., & Towsley, D. (2003). Modeling malware spreading dynamics. In *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No. 03CH37428)*, volume 3 (pp. 1869–1879).: IEEE.

Glazebrook, K. D., Ruiz-Hernandez, D., & Kirkbride, C. (2006). Some indexable families of restless bandit problems. *Adv. Appl. Probab.*, 38(3), 643–672.

Group, B. S. F. T. et al. (2002). The frequency of breast cancer screening: results from the ukcccr randomised trial. *European Journal of Cancer*, 38(11), 1458–1464.

Hethcote, H. W. (1997). An age-structured model for pertussis transmission. *Mathematical biosciences*, 145(2), 89–136.

Hoffmann, J. & Caramanis, C. (2018). The cost of uncertainty in curing epidemics. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 2(2), 31.

Holmes, K. K., Bertozzi, S., Bloom, B. R., & Jha, P. (2017). *Disease Control Priorities, (Volume 6): Major Infectious Diseases*. The World Bank.

Holte, R., Mok, A., Rosier, L., Tulchinsky, I., & Varvel, D. (1989). The pinwheel: A real-time scheduling problem. In *Proceedings of the 22nd Hawaii International Conference of System Science* (pp. 693–702).

Honda, J. & Takemura, A. (2010). An asymptotically optimal bandit algorithm for bounded support models. In *COLT* (pp. 67–79).: Citeseer.

Hsu, Y.-P. (2018). Age of information: Whittle index for scheduling stochastic arrivals. In *2018 IEEE ISIT* (pp. 2634–2638).: IEEE.

Irpan, A. (2018). Deep reinforcement learning doesn't work yet. https://www.alexirpan.com/2018/02/14/rl-hard.html.

Isella, L., Stehlé, J., Barrat, A., Cattuto, C., Pinton, J.-F., & Van den Broeck, W. (2011). What's in a crowd? analysis of face-to-face behavioral networks. *J. Theor. Biol.*, (pp. 166).

Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4, 237–285.

Kamarthi, H., Vijayan, P., Wilder, B., Ravindran, B., & Tambe, M. (2020). Influence maximization in unknown social networks: Learning policies for effective graph sampling. In *AAMAS* (pp. 575–583).

Kaufmann, E., Korda, N., & Munos, R. (2012). Thompson sampling: An asymptotically optimal finite-time analysis. In *International conference on algorithmic learning theory* (pp. 199–213).: Springer.

Kempe, D., Kleinberg, J., & Tardos, E. (2003). Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 137–146).: ACM.

Khalil, E., Dai, H., Zhang, Y., Dilkina, B., & Song, L. (2017). Learning combinatorial optimization algorithms over graphs. In *NeurIPS* (pp. 6348–6358).

Kipf, T. N. & Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In *ICLR-17* Toulon.

Kirkeby, C., Halasa, T., Gussmann, M., Toft, N., & Græsbøll, K. (2017). Methods for estimating disease transmission rates: Evaluating the precision of poisson regression and two novel methods. *Sci. Rep.*, 7, 9496.

Kleinberg, R. & Immorlica, N. (2018). Recharging bandits. In *2018 IEEE 59th FOCS* (pp. 309–319).: IEEE.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *NeurIPS* (pp. 1097–1105).

Kumar, U. D. & Saranga, H. (2010). Optimal selection of obsolescence mitigation strategies using a restless bandit model. *European Journal of Operational Research*, 200(1), 170–180.

Lanczos, C. (1950). *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*. United States Gov. Press Office Los Angeles, CA.

Lee, E., Lavieri, M. S., & Volk, M. (2019). Optimal screening for hepatocellular carcinoma: A restless bandit model. *Manufacturing & Service Operations Management*, 21(1), 198–212.

Leskovec, J. & Krevl, A. (2014). Stanford large network dataset collection. http://snap.stanford.edu/data.

Li, C., Wang, H., & Van Mieghem, P. (2012). Degree and principal eigenvectors in complex networks. In *ICRN* (pp. 149–160).: Springer.

Maghami, M. & Sukthankar, G. (2012). Identifying influential agents for advertising in multi-agent markets. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 2* (pp. 687–694).: International Foundation for Autonomous Agents and Multiagent Systems.

Maillard, O.-A., Munos, R., & Stoltz, G. (2011). A finite-time analysis of multi-armed bandits problems with kullback-leibler divergences. In *Proceedings of the 24th annual Conference On Learning Theory* (pp. 497–514).: JMLR Workshop and Conference Proceedings.

Malone, N. C., Williams, M. M., Smith Fawzi, M. C., Bennet, J., Hill, C., Katz, J. N., & Oriol, N. E. (2020). Mobile health clinics in the united states. *International journal for equity in health*, 19(1), 1–9.

Mansour, Y., Slivkins, A., & Syrgkanis, V. (2015). Bayesian incentive-compatible bandit exploration. In *Proceedings of the Sixteenth ACM Conference on Economics and Computation* (pp. 565–582).

Marsden, A. (2013). Eigenvalues of the laplacian and their relationship to the connectedness of a graph. *University of Chicago, REU*.

Mate, A., Killian, J. A., Xu, H., Perrault, A., & Tambe, M. (2020). Collapsing bandits and their application to public health intervention. In *2020 NeurIPS*.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., & Riedmiller, M. (2013). Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540), 529.

Murphy, S. A., Oslin, D. W., Rush, A. J., & Zhu, J. (2007). Methodological challenges in constructing effective treatment sequences for chronic psychiatric disorders. *Neuropsychopharmacology*, 32(2), 257–262.

Neke, N., Reifferscheid, A., Buchberger, B., & Wasem, J. (2018). Time and cost associated with utilization of services at mobile health clinics among pregnant women. *BMC health services research*, 18(1), 1–10.

Nino-Mora, J. (2001). Restless bandits, partial conservation laws and indexability. *Advances in Applied Probability*, (pp. 76–98).

Panzarasa, P., Opsahl, T., & Carley, K. M. (2009). Patterns and dynamics of users' behavior and interaction: Network analysis of an online community. *J. Assoc. Inf. Sci. Technol.*, (pp. 911).

Papadimitriou, C. H. & Tsitsiklis, J. N. (1994). The complexity of optimal queueing network control. In *Proceedings of IEEE 9th Annual Conference on Structure in Complexity Theory* (pp. 318–322).: IEEE.

Parr, R. & Russell, S. J. (1998). Reinforcement learning with hierarchies of machines. In *NeurIPS* (pp. 1043–1049).

Pohjosenperä, T., Kotavaara, O., & Juga, J. (2019). Mobile health care facilities vs. health centres–comparing the service structure strategies in reducing co2 emissions. -.

Prakash, B. A., Chakrabarti, D., Valler, N. C., Faloutsos, M., & Faloutsos, C. (2012). Threshold conditions for arbitrary cascade models on arbitrary networks. *Knowledge and information systems*, 33(3), 549–575.

Putri, W. C., Muscatello, D. J., Stockwell, M. S., & Newall, A. T. (2018). Economic burden of seasonal influenza in the united states. *Vaccine*, 36(27).

Qian, Y., Zhang, C., Krishnamachari, B., & Tambe, M. (2016). Restless poachers: Handling exploration-exploitation tradeoffs in security domains. In *In 2016 AAMAS* (pp. 123–131).

Qiu, W., Chen, H., & An, B. (2019). Dynamic electronic toll collection via multi-agent deep reinforcement learning with edge-based graph convolutional networks. In *IJCAI* (pp. 4568–4574).

Ren, Y., Jiang, M., Yao, Y., Wu, T., Wang, Z., Li, M., & Choo, K.-K. R. (2018). Node immunization in networks with uncertainty. In *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)* (pp. 1392–1397).: IEEE.

Rocha, L. E., Liljeros, F., & Holme, P. (2010). Information dynamics shape the sexual networks of internet-mediated prostitution. *Proceedings of the National Academy of Sciences*, 107(13), 5706–5711.

Ross, C. E. & Mirowsky, J. (2001). Neighborhood disadvantage, disorder, and health. *Journal of health and social behavior*, (pp. 258–276).

Saad-Roy, C., Shuai, Z., & van den Driessche, P. (2016). A mathematical model of syphilis transmission in an msm population. *Mathematical biosciences*, 277, 59–70.

Saha, S., Adiga, A., Prakash, B. A., & Vullikanti, A. K. S. (2015). Approximation algorithms for reducing the spectral radius to control epidemic spread. In *Proceedings of the 2015 SIAM International Conference on Data Mining* (pp. 568–576).: SIAM.

Salathé, M., Kazandjieva, M., Lee, J. W., Levis, P., Feldman, M. W., & Jones, J. H. (2010). A high-resolution human contact network for infectious disease transmission. *Proceedings of the National Academy of Sciences*, 107(51), 22020–22025.

Scaman, K., Kalogeratos, A., & Vayatis, N. (2016). Suppressing epidemics in networks using priority planning. *IEEE Transactions on Network Science and Engineering*.

Schwitters, A., Lederer, P., Zilversmit, L., Gudo, P. S., Ramiro, I., Cumba, L., Mahagaja, E., & Jobarteh, K. (2015). Barriers to health care in rural mozambique: a rapid ethnographic assessment of planned mobile health clinics for art. *Global Health: Science and Practice*, 3(1), 109–116.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. (2016). Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587), 484–489.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al. (2017). Mastering the game of go without human knowledge. *nature*, 550(7676), 354–359.

Smith, J. A. & Moody, J. (2013). Structural effects of network sampling coverage i: Nodes missing at random. *Social networks*, 35(4), 652–668.

Stephanie, W., Hill, C., Ricks, M. L., Bennet, J., & Oriol, N. E. (2017). The scope and impact of mobile health clinics in the united states: a literature review. *International journal for equity in health*, 16(1), 1–12.

Sun, C. & Hsieh, Y.-H. (2010). Global analysis of an seir model with varying population size and vaccination. *Applied Mathematical Modelling*, 34(10), 2685–2697.

Sutton, R. S. & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

Sutton, R. S. et al. (1998). *Introduction to reinforcement learning*, volume 135. MIT press Cambridge.

Sutton, R. S., Precup, D., & Singh, S. (1999). Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2), 181–211.

Swarup, S., Eubank, S. G., & Marathe, M. V. (2014). Computational epidemiology as a challenge domain for multiagent systems. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems* (pp. 1173–1176).: Citeseer.

Taylor-Robinson, D. (1994). Chlamydia trachomatis and sexually transmitted disease.

Tong, H., Prakash, B. A., Eliassi-Rad, T., Faloutsos, M., & Faloutsos, C. (2012). Gelling, and melting, large graphs by edge manipulation. In *Proceedings of the 21st ACM international conference on Information and knowledge management* (pp. 245–254).: ACM.

Tuberculosis, I. U. A. & Disease, L. (2018). Community-based active case finding in india can test and treat more people with tb in hard-to-reach tribal areas. *Union news report*.

Vanhems, P. et al. (2013). Estimating potential infection transmission routes in hospital wards using wearable proximity sensors. *PloS one*, 8, 73970.

Vazirani, V. V. (2013). *Approximation algorithms*. Springer Science & Business Media.

Waldorf, B. (2007). Measuring rurality. http://www.incontext.indiana.edu/2007/january/2.asp. Accessed: 2021-05-24.

Wang, H. (2002). A survey of maintenance policies of deteriorating systems. *European journal of operational research*, 139(3), 469–489.

Wang, N. (2005). *Modeling and analysis of massive social networks*. PhD thesis, UMD.

Wang, Y., Chakrabarti, D.and Wang, C., & Faloutsos, C. (2003). Epidemic spreading in real networks: An eigenvalue viewpoint. In *22nd International Symposium on Reliable Distributed Systems, 2003. Proceedings.* (pp. 25–34).: IEEE.

Watkins, C. J. & Dayan, P. (1992). Q-learning. *Machine learning*, 8(3-4), 279–292.

Weber, R. R. & Weiss, G. (1990). On an index policy for restless bandits. *J. Appl. Probab.*, 27(3), 637–648.

Weenig, M. W. & Midden, C. J. (1991). Communication network influences on information diffusion and persuasion. *Journal of personality and social psychology*.

Whittle, P. (1988). Restless bandits: Activity allocation in a changing world. *Journal of applied probability*, (pp. 287–298).

Wilder, B., Onasch-Vera, L., Hudson, J., Luna, J., Wilson, N., Petering, R., Woo, D., Tambe, M., & Rice, E. (2018). End-to-end influence maximization in the field. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems* (pp. 1414–1422).: International Foundation for Autonomous Agents and Multiagent Systems.

Wilder, B., Yadav, A., Immorlica, N., Rice, E., & Tambe, M. (2017). Uncharted but not uninfluenced: Influence maximization with an uncertain network. In *AAMAS*, volume 17 (pp. 1305–1313).

Willett, W. C., Koplan, J. P., Nugent, R., Dusenbury, C., Puska, P., & Gaziano, T. A. (2006). Prevention of chronic disease by means of diet and lifestyle changes. *Disease Control Priorities in Developing Countries. 2nd edition*.

Xu, L., Bondi, E., Fang, F., Perrault, A., Wang, K., & Tambe, M. (2021). Dual-mandate patrols: Multi-armed bandits for green security. In *In AAAI 2021*, volume 35 (pp. 14974–14982).

Yadav, A., Chan, H., Jiang, A. X., Xu, H., Rice, E., & Tambe, M. (2016a). Using social networks to aid homeless shelters: Dynamic influence maximization under uncertainty. In *AAMAS*, volume 16 (pp. 740–748).

Yadav, A., Kamar, E., Grosz, B., & Tambe, M. (2016b). Healer: Pomdp planning for scheduling interventions among homeless youth. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems* (pp. 1504–1506).: International Foundation for Autonomous Agents and Multiagent Systems.

Yi, J., Hsieh, C.-J., Varshney, K., Zhang, L., & Li, Y. (2017). Scalable demand-aware recommendation. *arXiv preprint arXiv:1702.06347*.

Zaremba, W., Sutskever, I., & Vinyals, O. (2014). Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*.

Zeng, C., Wang, Q., Mokhtari, S., & Li, T. (2016). Online context-aware recommendation with time varying multi-armed bandit. In *Proceedings of the 22nd ACM SIGKDD* (pp. 2025–2034).

Zhang, Y. & Prakash, B. A. (2015). Data-aware vaccine allocation over large networks. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 10(2), 20.