Near-Optimal Algorithms for Controlling Propagation at Group Scale on Networks

Yao Zhang, Abhijin Adiga, Sudip Saha, Anil Vullikanti, and B. Aditya Prakash

Abstract—Given a network with groups, such as a contact-network grouped by ages, which are the best groups to immunize to control the epidemic? Equivalently, how to choose best communities in social media like Facebook to stop rumors from spreading? Immunization is an important problem in multiple different domains like epidemiology, public health, cyber security, and social media. Additionally, clearly immunization at group scale (like schools and communities) is more realistic due to constraints in implementations and compliance (e.g., it is hard to ensure specific individuals take the adequate vaccine). Hence, efficient algorithms for such a "group-based" problem can help public-health experts take more practical decisions. However, most prior work has looked into individual-scale immunization. In this paper, we study the problem of controlling propagation at group scale. We formulate a set of novel Group Immunization problems for multiple natural settings (for both threshold and cascade-based contagion models under both node-level and edge-level interventions) and develop multiple efficient algorithms, including provably approximate solutions. Finally, we show the effectiveness of our methods via extensive experiments on real and synthetic datasets.

Index Terms—Graph mining, social networks, immunization, diffusion, groups

1 Introduction

Infectious diseases account for a large fraction of deaths worldwide. The main public health response to containing epidemic outbreaks is by vaccination and social distancing, e.g., [1], [2]. These interventions have resource constraints (e.g., limited supply of vaccines and the high cost of social distancing), and therefore, designing optimal control strategies is an active area of research in public health policy planning, e.g., [1], [3], [4], [5], [6], [7]. However, optimal strategies based on node level characteristics, such as the degree or spectral properties [6], [7] cannot be easily turned into implementable policies, because such targeted immunization of specific individuals raises significant social and moral issues. As a result, vaccination policies, such as those specified by CDC are at the level of groups (e.g., based on demographics), and almost all the efforts in epidemiology are focused on developing group level strategies, even though this may lead to sub-optimal solutions compared to the individual level policies. For instance, Medlock et al. [1] develop an optimal vaccine allocation for different age groups. Even so, all prior work on optimal group level immunization has focused on differential equation based models, and has not been studied on network models of epidemic spread. Implementing such interventions is challenging because people "comply" with them based on their individual utility. We model such limited compliance by random vaccine allocation within each group, which motivates our paper. Our focus in this paper is on developing interventions that can be implemented before the start of the epidemic. Further, interventions can be of two kinds: vaccination (which can be modeled in terms of node removals) and social distancing (which can be modeled in terms of edge removal, e.g., reducing contacts between certain sub-populations). We consider two kinds of metrics: (1) maximizing the expected number of people who do not get infected, and (2) minimizing the time for the epidemic to die out. These are both commonly studied metrics in public health (see, e.g., [8], [9]). Most of the work in mathematical epidemiology has been formalized in terms of reducing the reproductive number. However, these methods do not extend to network based models. In this paper, we develop algorithms for optimizing these metrics in two different models of diffusion.

Similar diffusion processes arise in other domains such as social media, e.g., the spread of spam/rumors on Facebook, Twitter, LiveJournal or Friendster. These are also commonly modeled by models such as the Linear Threshold (LT) model [10]. Analogous to the public-health case, we can control such processes by 'immunization' via blocking users or preventing some interactions, such that the expected number of users who adopt spam/rumors is minimal. Past work has studied individual-level based immunization algorithms for the LT model [11]. However, it is more realistic to issue a warning bulletin on group pages, and some members within those groups comply with the warning to stop disseminating rumors. Similarly, Twitter can warn a group of accounts to control the spread of the malicious tweets. The same holds true for user groups in Friendster and LiveJournal.

In this paper, we present a unified approach to study strategies for controlling the spread of diffusion processes through group level interventions, capturing both uncertainty and lack of control at high resolution within groups. The main contributions of our paper are:

- *Y. Zhang and B.A. Prakash are with the Department of Computer Science, Virginia Tech, Blacksburg, VA 24061. E-mail: {yaozhang, badityap}@cs.vt.edu.*
- *A. Adiga is with NDSSL, Biocomplexity Institute of Virginia Tech, Blacksburg, VA 24061. E-mail: abhijin@vbi.vt.edu.*
- *S. Saha and A. Vullikanti are with the Department of Computer Science, Virginia Tech, Blacksburg, VA 24061, and NDSSL, Biocomplexity Institute of Virginia Tech, Blacksburg, VA 24061. E-mail: {ssaha, akumar}@vbi.vt.edu.*

Manuscript received 29 Dec. 2015; revised 9 July 2016; accepted 24 Aug. 2016. Date of publication 1 Sept. 2016; date of current version 2 Nov. 2016. Recommended for acceptance by A. Gionis.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TKDE.2016.2605088

1) *Problem Formulation*: We develop group level intervention problems in both the LT model, and the SIS/SIR models, for which we consider a spectral radius based formulation. We consider arbitrarily specified groups, and interventions that involve both edge and node removal, modeling quarantining and vaccination, respectively. The interventions specify the number x_i of nodes/edges that can be removed within each group C_i; however, these are chosen randomly within the group. These problems generalize the node level problems and have not been studied before.
2) *Effective Algorithms*: We develop efficient theoretical and practical algorithms for the four problem classes we consider, including provable approximation algorithms (SDP and GroupGreedyWalk). We find that diverse kinds of techniques are needed for these problems—submodular function maximization on an integer lattice, semidefinite programming, quadratic programming, and the link between closed walks and spectral radius. Our algorithms also leverage prior techniques for analyzing contagion processes, e.g., [6], [12], [13], but require non-trivial extensions.
3) *Experimental Evaluation*: We present extensive experiments on multiple real datasets including epidemiological and social networks, and demonstrate that our algorithms outperform other competitors on node and edge deletion at group scale for controlling infection as well as spectral radius minimization.

Outline of the Paper. The rest of the paper is organized as follows. We first discuss the related work in Section 2, and then formulate the Group Immunization Problem in Section 3. Section 4 presents our algorithms for different settings of the problem for both edge and node removal. Experimental results on several datasets are in Section 5. We finally discuss future work, and conclude in Section 6.

2 Related Work

In general, there has been a lot of interest in studying dynamical processes on large graphs like (a) blogs and propagations [14], [15], (b) information cascades [16], [17]; (c) marketing and product penetration [18], [19] and (d) malware prediction [20]. These dynamic processes are all closely related to virus propagation in epidemiology, rumor spread in social media, malware outbreaks in computer networks, etc. In this section, we review related work mainly from four areas: epidemiology, propagation models, immunization and other diffusion based optimization problems. In short, past work concentrates on *individual-based* immunization—in contrast, in this paper we study group-based immunization problems under various models.

Epidemiology. The classical texts on epidemic models and analysis are May and Anderson [8] and Hethcote [21]. Widely studied epidemiological models include *homogeneous models* [8], [9], [22], which assume that every individual has equal contact with others in the population.

Propagation Models. There are broadly two types of propagation models which have been used to describe dynamical processes on graphs: threshold based and cascade style.

Threshold based models are well-motivated in the social science literature [23], [24], [25] to represent 'threshold' behaviors, e.g., ideas/spam/rumors on Twitter and Facebook. A classic example is the linear threshold model, which has been extensively studied [10]. In this paper, we study the problem of minimizing propagation for the LT model.

Cascade style models, such as the 'flu-like' Susceptible-Infectious-Susceptible (SIS), 'mumps-like' Susceptible-Infectious-Recovered (SIR) and its special-case the Independent Cascade (IC) [8], [9], [10], [22], are popular in epidemiology literature to model different epidemiological states of people, and their state-transitions. Much work has gone into in finding the 'epidemic threshold' for such models (the minimum virulence of a virus which results in an epidemic over the network). For example, recent studies [12], [26] show that the spectral radius of the underlying network (the largest eigenvalue of the adjacency matrix of the graph) is related to the epidemic threshold for a wide-range of cascade models. Hence, here we investigate how to control an epidemic by minimizing the spectral radius for cascade style models.

Immunization. There has been much work on finding optimal strategies for vaccination and social distancing [1], [3], [4], [5], [6], [7]. Much of the work in the epidemiology literature has been based on differential equation methods [1], [3], [4]. Cohen et al. [5] studied the popular *acquaintance* immunization policy (pick a random person, and immunize one of its neighbors at random). Using game theory, Aspnes et al. [27] developed inoculation strategies for victims of viruses under random starting points. Kuhlman et al. [28] studied two formulations of the problem of blocking a contagion through edge removals under the model of discrete dynamical systems. Tong et al. [7], 29], Van Miegham et al. [30], Prakash et al. [6] proposed various node-based and edge-based immunization algorithms based on minimizing the largest eigenvalue of the graph. Other non-spectral approaches for immunization have been studied by Budak et al. [31], He et al. [32], Khalil et al. [11], Saha et al. [13], and Zhang et al. [33]. All of these papers studied individual-based immunization (where either one targets specific individuals or whole demographics). Here we study group-based problems, where vaccines are distributed randomly inside groups.

Other Diffusion Problems. Other diffusion based optimization problems include the influence maximization problem, which was introduced by Domingos and Richardson [34], and formulated by Kempe et al. [10] as a combinatorial optimization problem. They proved it is NP-Hard and also gave a simple $(1 - 1/e)$-approximation based on the submodularity of expected spread of a set of starting seeds. Recently the paper by Eftekhar et al. [35] studied this problem at group scale. Other such problems where we wish to select a subset of '*important*' vertices on graphs, include 'outbreak detection' [36] and 'finding most-likely culprits of epidemics' [37]. Purohit et al. [38] looked into 'zooming-out' of a graph by forming groups based on similar influence.

3 Our Problem Formulations

Table 1 lists the main symbols we use throughout the paper. Here we assume our graph $G(V, E)$ is directed and weighted. We refer to both node and edge level interventions as immunization.

Groups in a Graph. For a graph $G(V, E)$, we assume that the edge (node) set is partitioned into groups $C = \{C_1, \ldots, C_n\}$

TABLE 1
Terms and Symbols

Symbol	Definition and Description
$G(V, E)$	graph G with the node set V and the edge set E
C	set containing groups
A	set of initial infected nodes
n	the number of groups in the graph
m	budget (the number of vaccines)
p_{uv}	weight on edge $e(u, v)$
$g(v)$	group index of node v, i.e., $g(v) = i$ if $v \in C_i$
$g(u, v)$	group index of edge (u, v), i.e., $g(u, v) = i$ if $(u, v) \in C_i$
$\mathbf{x}$	vaccine allocation vector $(x_1, \ldots, x_n)$ for edges/nodes
$\sigma_{C,A}(\mathbf{x})$	the expected number of infected nodes at the end when $\mathbf{x}$ is allocated to edges
$\sigma'_{C,A}(\mathbf{x})$	the expected number of infected nodes at the end when $\mathbf{x}$ is allocated to nodes
$\mathbf{e}_k$	vector with $e_k = 1$ and $e_i = 0$ for $i \neq k$
$\mathbf{M}_{\mathbb{E}}(\mathbf{x})$	$\mathbb{E}[\mathbf{M}(\mathbf{x})]$
$\Delta_{\mathbb{E}}(\mathbf{x})$	maximum expected degree of $G(\mathbf{x})$
$\lambda_{\mathbb{E}}(\mathbf{x})$	expected spectral radius of $\mathbf{M}(\mathbf{x})$
$\lambda(\mathbf{M}_{\mathbb{E}}(\mathbf{x}))$	spectral radius of the expected matrix $\mathbf{M}_{\mathbb{E}}(\mathbf{x})$
$\lambda_{\mathbb{E}}^{\min}$	minimum expected spectral radius over all $\mathbf{M}(\mathbf{x})$, i.e., $\min_x \lambda_{\mathbb{E}}(\mathbf{x})$
$\mathbf{x}_{\min}$	the allocation vector which minimizes $\lambda(\mathbf{M}_{\mathbb{E}}(\mathbf{x}))$ over all $\mathbf{x}$, i.e., $arg\,min_{\mathbf{x}} \lambda(\mathbf{M}_{\mathbb{E}}(\mathbf{x}))$
s	number of samples in GroupGreedyWalk

for the edge (node) immunization problems. For a node partition, C may correspond to groups of communities, locations, demographics, etc. And edge groups can be induced from node groups. For example, for an edge $e = (u, v)$, if u and v belong to a group C_t, then $e \in C_t$, otherwise it belongs to group $C_{ij} = \{e_{uv} | u \in C_i, v \in C_j\}$. The edge groups we defined ensure that every edge e has a group even if the endpoints of e belong to different node groups.Note that we assume there are no overlaps among groups.

Allocating Vaccines to Groups. We define $\mathbf{x} = (x_1, \ldots, x_n)$ as the vaccine allocation vector, i.e., if we give x_i vaccines to group C_i, x_i edges (nodes) will be uniformly randomly removed from C_i, which means those edges/nodes will not be involved in the diffusion process. The objective of our immunization problem is to find an allocation that controls the diffusion process most effectively.

For edge deletion, a good solution tends to give more vaccines to the edge groups where edges inside have high chance to be a part of cuts/walks. Similarly, for node deletion, we prefer the node groups where nodes insides have high impact on the influence/eigenvalue.

Main Idea of Our Problem Definitions. We give two different sets of problems which cover a wide range of contagion-like processes both threshold-based and cascade-style in the next two sections. In addition, all our problems have been carefully formulated to be seamless generalizations of the corresponding individual-level problems.

3.1 Problem Definition under LT Model

Our first set of problems are based on the LT model which is a well-known model for social media and complex propagations [10] suited for representing 'threshold' behaviors for activation. As mentioned in the introduction, the vaccination problem here can help to control such processes like spam and rumors on Twitter and Facebook. Under the LT model, our goal is to minimize the expected number of infected nodes at the end of diffusion, in other words, maximize the expected number of nodes we can save from being infected, by selecting groups for removing edges/nodes.

In the LT model, a node v can be influenced by each neighbor u according to a weight p_{uv} where $\sum_{e(u,v)\in E} p_{uv} \leq 1$. The diffusion process proceeds as follows: at the start, every node u uniformly randomly chooses a threshold θ_u from the range [0,1], which represents the weighted fraction of u's neighbors that must be active to activate u; an inactive node u becomes active at time $t+1$ if $\sum_{w\in N_i^t} p_{wu} \geq \theta_u$ where N_u^t is the set of active neighbors of u at time t; all active nodes will stay active. The process stops when no additional node becomes active. Each group may have some seeds (initial infected nodes). The seeds will spread information/virus by the LT model.

For the edge deletion under the LT model, let $\sigma_{C,A}(\mathbf{x})$ ($\mathbb{Z}^n \rightarrow \mathbb{R}$) denote the expected number of infected nodes in G (the footprint of G), given seed set A and vaccine allocation vector $\mathbf{x}$ for the group set C. Now we are ready to define the edge version of the problem under the LT model.

Problem 1: Group Immunization under LT model (edge version):

Given: *Graph $G(V, E)$, a partition of the edge set $C = \{C_1, \ldots, C_n\}$, seed set A and m vaccines (budget). Let $\mathbf{x}$ be the edge vaccine allocation vector.*

Find: *The optimum allocation $\mathbf{x}_{\text{opt}}$ which maximizes $f(\mathbf{x}) = \sigma_{C,A}(\mathbf{0}) - \sigma_{C,A}(\mathbf{x})$ s.t. $|\mathbf{x}| \leq m$.*

Next, we define the node version of this problem. Let $\sigma'_{C,A}(\mathbf{x})$ denote the footprint of G. It is same as $\sigma_{C,A}(\mathbf{x})$ except that the allocation vector $\mathbf{x}$ corresponds to node vaccination.

Problem 2: Group Immunization under LT model (node version):

Given: *Graph $G(V, E)$, a partition of the vertex set $C = \{C_1, \ldots, C_n\}$, seed set A and m vaccines (budget). Let $\mathbf{x}$ be the node vaccine allocation vector.*

Find: *The optimum allocation $\mathbf{x}_{\text{opt}}$ which maximizes $f'(\mathbf{x}) = \sigma'_{C,A}(\mathbf{0}) - \sigma'_{C,A}(\mathbf{x})$ s.t. $|\mathbf{x}| \leq m$.*

Hardness of Our Problems. Problems 1 and 2 are NP-hard as their special case, individual-level based immunizations (when each edge/node is a group), are NP-hard themselves [11], [33].

3.2 Problem Definition for Spectral Radius

Our second set of problems are based on the spectral radius formulation [7], [29] for a variety of cascade models including the fundamental SIR ('mumps-like' which generalizes the well-known IC model [10]), SIS ('flu-like'), and SEIS (with incubation period) models. In the SIS/SIR models, every node can be either susceptible (S), infectious (I) or recovered (R). Each infected node u (in state I) can infect each susceptible neighbor v (in state S) with the probability p_{uv}. In the SIS model, each infected node u can switch to the susceptible state with the recovery rate δ. In the SIR model, each infected node u can switch to the recovered state with the recovery rate δ, meaning u cannot be infected again.

Spectral radius, denoted by λ, refers to the largest eigenvalue of the adjacency matrix of a graph G. Recent results [12], [26] have shown that λ is connected to the reproduction number in epidemiology, and determines the phase-transition ('epidemic threshold' τ) between epidemic/non-epidemic regimes in a very large range of cascade-style models [12], including SIR, SIS , SEIS and so on. As shown in [12], $\tau \propto \lambda$,

and if $\tau < 1$ the disease will die out quickly irrespective of initial conditions. This gives us the motivation to control the disease spread by minimizing λ in the underlying network.

Tong et al. [7], [29] proposed effective node-based and edge-based individual immunization methods to minimize λ. Following their methodology, in this paper we aim to maximize the drop of the spectral radius of G, $\Delta\lambda$, when vaccines are allocated to groups. Similar to Problems 1 and 2, when x_i vaccines are given to group C_i, we uniformly remove x_i nodes/edge at random. Hence, we want to find the optimal allocation $\mathbf{x}$ such that the expectation of $\Delta\lambda$, $\mathbb{E}[\Delta\lambda](\mathbf{x})$ is maximum. Note that we do not define the problems here based on the 'footprint' (as in the previous section for LT) for primarily two reasons: (a) these versions naturally generalize the corresponding individual-level immunization problems studied in past literature [7], [29]; and (b) due to the epidemic threshold results, using the spectral radius allows us to immediately formulate a general problem for multiple cascade-style models (like SIR/SIS/IC) each with differences in their exact spreading process which we can ignore. Formally our problems are:

PROBLEM 3: GROUP IMMUNIZATION for spectral radius (edge version)

GIVEN: *Graph $G(V, E)$, a partition of the edge set $C = \{C_1, \ldots, C_n\}$, and m vaccines (budget). Let $\mathbf{x}$ be the edge vaccine allocation vector, and let $\mathbb{E}[\Delta\lambda](\mathbf{x})$ denote the expected drop in the spectral radius after the immunization.*

FIND: *The optimum allocation $\mathbf{x}_{\text{opt}}$ which maximizes $\mathbb{E}[\Delta\lambda]$, i.e., $\mathbf{x}_{\text{opt}} = \arg\max_{\mathbf{x}} \mathbb{E}[\Delta\lambda](\mathbf{x})$ s.t. $|\mathbf{x}| \leq m$.*

PROBLEM 4: GROUP IMMUNIZATION for spectral radius (node version)

GIVEN: *Graph $G(V, E)$, a partition of the node set $C = \{C_1, \ldots, C_n\}$, and m vaccines (budget). Let $\mathbf{x}$ be the edge vaccine allocation vector, and let $\mathbb{E}[\Delta\lambda](\mathbf{x})$ denote the expected drop in the spectral radius after the immunization.*

FIND: *The optimum allocation $\mathbf{x}_{\text{opt}}$ which maximizes $\mathbb{E}[\Delta\lambda]$, i.e., $\mathbf{x}_{\text{opt}} = \arg\max_{\mathbf{x}} \mathbb{E}[\Delta\lambda](\mathbf{x})$ s.t. $|\mathbf{x}| \leq m$.*

Hardness of Our Problems. Problems 3 and 4 are NP-hard too—their special cases, individual-level immunizations are NP-hard [7], [29].

4 PROPOSED METHODS

We first discuss our algorithms for the GROUP IMMUNIZATION problem under the LT model (Sections 4.1 and 4.2 for Problems 1 and 2), and then the spectral radius versions (Sections 4.3 and 4.4 for Problems 3 and 4).

4.1 Edge Deletion under LT Model

Recall that the function $f(\mathbf{x})$ in Problem 1 is not a simple set function; it is over an *integer lattice*. Hence the submodularity property used in [11] is not applicable to our problem, and we can not simply apply their greedy algorithm. Instead, our approach is to carefully identify a 'submodularity like' condition that is satisfied by our function $f(\mathbf{x})$, for which a greedy algorithm gives good performance. Let $\mathbf{e}_k$ be the vector with 1 at the kth index and $\mathbf{0}$ be the all zeros vector. We consider the following three properties.

- (P_1) $f(\mathbf{x}) \geq 0$ and $f(\mathbf{0}) = 0$.
- (P_2) (Non-decreasing) $f(\mathbf{x}) \leq f(\mathbf{x} + \mathbf{e}_k)$ for any k.
- (P_3) (Diminishing returns) For any $\mathbf{x}' \geq \mathbf{x}$ and k, we have $f(\mathbf{x} + \mathbf{e}_k) - f(\mathbf{x}) \geq f(\mathbf{x}' + \mathbf{e}_k) - f(\mathbf{x}')$.

The notion of submodularity of set functions has been extended to functions over integer lattices—see, e.g., [39], which shows that a greedy algorithm gives a constant factor approximation to submodular lattice functions with budget constraints. We note that in the context of functions defined on an integer lattice, unlike in the case of set functions, submodularity need not be equivalent to the diminishing return property. Besides, there are multiple non-equivalent definitions of the diminishing return property, as observed in [39]. Next, we show that in Theorem 1 that a greedy algorithm gives an $(1 - 1/e)$-factor approximation to an integer lattice function satisfying the properties (P_1), (P_2) and (P_3) above, and our objective function follows all above properties (Lemma 2). Note that it is not clear whether the analysis of [39] implies a similar bound for the kind of functions $f(\mathbf{x})$ we need to consider here.

Lemma 1. *Suppose $\mathbf{y} = (y_i, \ldots, y_n)^T$ where $y_i \in \mathbb{Z}^*$ and $\sum_j y_j = m$, then $f(\mathbf{x} + \mathbf{y}) - f(\mathbf{x}) \leq \sum_j y_j (f(\mathbf{x} + \mathbf{e}_j) - f(\mathbf{x}))$.*

Proof. The proof is in the appendix.[1] □

Theorem 1. *Suppose $f(\mathbf{x})$, $\mathbf{x} \in \mathbb{Z}^n$ satisfies the properties (P_1), (P_2) and (P_3) above. Then, Algorithm 1 gives a $(1 - 1/e)$-approximate solution to the problem of maximizing $f(\mathbf{x})$ subject to $\sum_i x_i \leq m$.*

Proof. Suppose $\mathbf{x}$ is the solution from the greedy algorithm, and $\mathbf{x}^*$ is the optimal solution. Hence, we have $\sum_j x_j = \sum_j x_j^* = m$. Since $\sigma_C(\mathbf{0})$ is constant, the greedy algorithm is equivalent to

$$C^* = \arg\max_{C_i} f(\mathbf{x} + \mathbf{e}_i) - f(\mathbf{x}).$$

Let us define $\mathbf{x}^{(i)}$ as the solution got from the ith iteration of the greedy algorithm, hence $\mathbf{x} = \mathbf{x}^{(m)}$. And $\mathbf{x}^*$ can be represent as $\sum_j x_j^* \mathbf{e}_j$. We have

$$\begin{aligned} f(\mathbf{x}^*) &\leq f(\mathbf{x}^* + \mathbf{x}^{(i)}) \\ &= f(\mathbf{x}^{(i)}) + (f(\mathbf{x}^* + \mathbf{x}^{(i)}) - f(\mathbf{x}^{(i)})) \\ &\leq f(\mathbf{x}^{(i)}) + \sum_j x_j^* (f(\mathbf{x}^{(i)} + \mathbf{e}_j) - f(\mathbf{x}^{(i)})) \text{ (\textbf{Lemma 1})} \\ &\leq f(\mathbf{x}^{(i)}) + \sum_j x_j^* (f(\mathbf{x}^{(i+1)}) - f(\mathbf{x}^{(i)})) \text{ (\textbf{Greedy Alg.})} \\ &= f(\mathbf{x}^{(i)}) + m(f(\mathbf{x}^{(i+1)}) - f(\mathbf{x}^{(i)})). \end{aligned}$$

Hence, $f(\mathbf{x}^{(i+1)}) \geq (1 - \frac{1}{m}) f(\mathbf{x}^{(i)}) + \frac{1}{m} f(\mathbf{x}^*)$. Recursively, we can get $f(\mathbf{x}^{(i)}) \geq (1 - (1 - \frac{1}{m})^i) f(\mathbf{x}^*)$. Therefore, $f(\mathbf{x}) = f(\mathbf{x}^{(m)}) \geq (1 - (1 - \frac{1}{m})^m) f(\mathbf{x}^*) \geq (1 - 1/e) f(\mathbf{x}^*)$. □

Algorithm 1. Greedy Algorithm

Require: f, budget m
1: $\mathbf{x} = 0$
2: **for** $j = 1$ to m **do**
3: $\quad i = \arg\max_{k=1,\ldots,n} f(\mathbf{x} + \mathbf{e}_k) - f(\mathbf{x})$
4: $\quad \mathbf{x} = \mathbf{x} + \mathbf{e}_i$
5: **end for**
6: **return x**

Now, we will show that the objective function $f(\mathbf{x}) = \sigma_{C,A}(\mathbf{0}) - \sigma_{C,A}(\mathbf{x})$ for the edge deletion problem under

1. The proof is in the appendix, which can be found at: http://people.cs.vt.edu/yaozhang/group-immu/

the LT model satisfies the properties stated in Theorem 1. In the ensuing discussion, we will assume without loss of generality that there is only one seed node. This is because, if there are multiple seed nodes, then, we can merge all of them to a single 'super' node (say s) in the following manner: for every vertex $v \in V \setminus A$, set $p_{sv} = \sum_{u \in N(v) \cap A} p_{uv}$, where $N(v)$ is the set of neighbors of v. We note that after this modification the edges between v and its susceptible neighbors are unchanged, and at time 0, $\sum_{w \in N_v} p_{uv} = \sum_{w \in N(v) \cap A} p_{wv} = p_{sv}$. Hence, $\sigma_{C,A}(\mathbf{x}) = \sigma_{C,s}(\mathbf{x})$. Henceforth, we will assume that there is only one seed node, and drop the subscript A from $\sigma_{C,A}(\mathbf{x})$, denoting it by $\sigma_C(\mathbf{x})$.

Lemma 2. *The function $f(\mathbf{x}) = \sigma_C(\mathbf{0}) - \sigma_C(\mathbf{x})$ satisfies the properties (P_1), (P_2) and (P_3) above.*

Proof. Property 1 is trivially true because, when $\mathbf{x} = \mathbf{0}$, by definition, $f(\mathbf{0}) = 0$, and since vaccination does not increase the number of infections, $\sigma_C(\mathbf{x}) \leq \sigma_C(\mathbf{0})$. For the rest of the proof, since $\sigma_C(\mathbf{0})$ is a constant, we only need to analyze $\sigma_C(\mathbf{x})$. Note that for any $\mathbf{x}' \geq \mathbf{x}$, we can find a sequence of vectors $(\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_l)$ for some l such that $\mathbf{x} = \mathbf{z}_1$, $\mathbf{x}' = \mathbf{z}_l$ and $\mathbf{z}_i = \mathbf{z}_{i-1} + \mathbf{e}_{k_{i-1}}$ for some index k_{i-1}. Therefore, it is enough to prove that Properties 1 and 2 hold for $\mathbf{x}' = \mathbf{x} + \mathbf{e}_j$ for some index j. Also, we can assume that $x_j < |C_j|$, for $j = 1, \ldots, n$, for if this is not true for some j, then, it implies that all the edges in C_j will be vaccinated, and therefore, we can simply remove all C_j from the analysis and reduce the budget by x_j.

Let $\mathcal{R}(\mathbf{x}) \subseteq 2^V$ be the collection of sets R satisfying $|R \cap C_i| = x_i$. Following the equivalence between influence in the LT model and the directed percolation process [10], we have $\sigma_C(\mathbf{x}) = \sum_{\hat{G}} \Pr[\hat{G}] \sum_{R \in \mathcal{R}(\mathbf{x})} \Pr[R] \gamma_C(\hat{G}, R)$, where the first sum is over all possible live-edge subgraphs $\hat{G}$ of G in the percolation process, $\Pr[R]$ is the probability when the set R is removed , and $\gamma_C(\hat{G}, R)$ is the expected number of infected nodes in $\hat{G}$ at the end of the LT process after the set R is removed. This can be rewritten as $\sigma_C(\mathbf{x}) = \sum_{\hat{G}} \Pr[\hat{G}] \sigma_C(\hat{G}, \mathbf{x})$, where $\Pr[\hat{G}]$ is the probability of sampling $\hat{G}$, and $\sigma_C(\hat{G}, \mathbf{x}) = \sum_{R \in \mathcal{R}(\mathbf{x})} \Pr[R] \gamma_C(\hat{G}, R)$. Henceforth, we will abbreviate $\gamma_C(\hat{G}, R)$ as $\hat{\gamma}(R)$.

We will show that $\sigma_C(\hat{G}, \mathbf{x})$ is non-increasing, i.e., $\sigma_C(\hat{G}, \mathbf{x}) \geq \sigma_C(\hat{G}, \mathbf{x}')$ where $\mathbf{x}' = \mathbf{x} + \mathbf{e}_j$, thereby showing that $f(\mathbf{x})$ satisfies Property 2. Since the number of nodes reachable from the seed node with R removed is at least as many as those with $R \cup \{e\}$ removed, for any $e \in C_j \setminus R$, we have $\hat{\gamma}(R) \geq \hat{\gamma}(R \cup \{e\})$. Therefore,

$$\begin{aligned}
\sigma_C(\hat{G}, \mathbf{x}') &= \sum_{R' \in \mathcal{R}(\mathbf{x}')} \Pr[R'] \hat{\gamma}(R') \\
&= \sum_{R \in \mathcal{R}(\mathbf{x})} \sum_{e \in C_j \setminus R} \frac{1}{|C_j| - x_j} \Pr[R] \hat{\gamma}(R \cup \{e\}) \\
&\leq \sum_{R \in \mathcal{R}(\mathbf{x})} \sum_{e \in C_j \setminus R} \frac{1}{|C_j| - x_j} \Pr[R] \hat{\gamma}(R) \\
&= \sum_{R \in \mathcal{R}(\mathbf{x})} \Pr[R] \hat{\gamma}(R) = \sigma_C(\hat{G}, \mathbf{x}).
\end{aligned}$$

Finally, we will show that $\sigma_C(\hat{G}, \mathbf{x} + \mathbf{e}_k) - \sigma_C(\hat{G}, \mathbf{x}) \leq \sigma_C(\hat{G}, \mathbf{x}' + \mathbf{e}_k) - \sigma_C(\hat{G}, \mathbf{x}')$. From the above discussion, this will imply that $f(\mathbf{x})$ satisfies Property 3. Suppose $\mathbf{x}' = \mathbf{x} + \mathbf{e}_j$, we have two cases to consider: (1). $\mathbf{e}_k = \mathbf{e}_j$; (2). $\mathbf{e}_k \neq \mathbf{e}_j$.

For $1 \leq i \leq n$, let $c_i = |C_i|$ and x_i denote the ith element in $\mathbf{x}$.

First, we consider case (1) ($\mathbf{e}_k = \mathbf{e}_j$). For $R \in \mathcal{R}(\mathbf{x})$, $\Pr[R] = \prod_i \frac{1}{\binom{c_i}{x_i}} = \rho \frac{1}{\binom{c_k}{x_k}}$, where, $\rho = \prod_{i \neq k} \frac{1}{\binom{c_i}{x_i}}$

$$\begin{aligned}
&\sigma_C(\hat{G}, \mathbf{x}) - \sigma_C(\hat{G}, \mathbf{x} + \mathbf{e}_k) \\
&\quad = \sum_{R \in \mathcal{R}(\mathbf{x})} \rho \frac{1}{\binom{c_k}{x_k}} \hat{\gamma}(R) - \sum_{R' \in \mathcal{R}(\mathbf{x}')} \rho \frac{1}{\binom{c_k}{(x_k+1)}} \hat{\gamma}(R') \\
&\quad = \rho \sum_{R \in \mathcal{R}(\mathbf{x})} \left[\frac{1}{\binom{c_k}{x_k}} \hat{\gamma}(R) - \frac{1}{x_k + 1} \sum_{e \in C_k \setminus R} \frac{1}{\binom{c_k}{x_k+1}} \hat{\gamma}(R \cup \{e\}) \right].
\end{aligned}$$

The factor $\frac{1}{x_k+1}$ is due to the fact that $R \cup \{e\}$ comes up in $(x_k + 1)$ combinations involving R and e. This simplifies to

$$\begin{aligned}
&\sigma_C(\hat{G}, \mathbf{x}) - \sigma_C(\hat{G}, \mathbf{x} + \mathbf{e}_k) \\
&\quad = \frac{\rho x_k! (c_k - x_k - 1)!}{c_k!} \sum_{R \in \mathcal{R}(\mathbf{x})} \sum_{e \in C_k \setminus R} \hat{\gamma}(R) - \hat{\gamma}(R \cup \{e\}). \quad (1)
\end{aligned}$$

Similarly, we have

$$\begin{aligned}
&\sigma_C(\hat{G}, \mathbf{x}') - \sigma_C(\hat{G}, \mathbf{x}' + \mathbf{e}_k) \\
&= \frac{\rho (x_k + 1)! (c_k - x_k - 2)!}{c_k!} \sum_{R' \in \mathcal{R}(\mathbf{x}')} \sum_{e \in C_k \setminus R'} \hat{\gamma}(R') - \hat{\gamma}(R' \cup \{e\}) \\
&= \frac{\rho (x_k + 1)! (c_k - x_k - 2)!}{c_k!} \sum_{R \in \mathcal{R}(\mathbf{x})} \frac{1}{(x_k + 1)} \sum_{e' \in C_k \setminus R} \\
&\qquad \sum_{e \in C_k \setminus (R \cup \{e'\})} \hat{\gamma}(R \cup \{e'\}) - \hat{\gamma}(R \cup \{e, e'\}).
\end{aligned}$$

From [11, proof of Theorem 6], $\hat{\gamma}(R) - \hat{\gamma}(R \cup \{e\}) \geq \hat{\gamma}(R \cup \{e'\}) - \hat{\gamma}(R \cup \{e, e'\})$ (supermodularity). Therefore, $(c_k - x_k - 1) \sum_e [\hat{\gamma}(R) - \hat{\gamma}(R \cup \{e\})] \geq \sum_{e'} \sum_e \hat{\gamma}(R \cup \{e'\}) - \hat{\gamma}(R \cup \{e, e'\})$. Hence proved.

Now, we consider case (2). Let

$$\Pr[R] = \rho' \frac{1}{\binom{c_k}{x_k}} \frac{1}{\binom{c_j}{x_j}},$$

where $\rho' = \prod_{i \neq j,k} \frac{1}{\binom{c_i}{x_i}}$. We can get $\sigma_C(\hat{G}, \mathbf{x}) - \sigma_C(\hat{G}, \mathbf{x} + \mathbf{e}_k)$ from Eqn. (1). And

$$\begin{aligned}
&\sigma_C(\hat{G}, \mathbf{x}') - \sigma_C(\hat{G}, \mathbf{x}' + \mathbf{e}_k) \\
&= \frac{\rho' x_k! (c_k - x_k - 1)!}{\binom{c_j}{(x_j+1)} c_k!} \sum_{R' \in \mathcal{R}(\mathbf{x}')} \sum_{e \in C_k \setminus R'} [\hat{\gamma}(R') - \hat{\gamma}(R' \cup \{e\})] \\
&= \frac{\rho' (x_j + 1)! (c_j - x_j - 1)!}{c_j!} \frac{x_k! (c_k - x_k - 1)!}{c_k!} \sum_R \frac{1}{x_j + 1} \\
&\qquad \sum_{e_j \in C_j \setminus R} \sum_{e \in C_k \setminus (R \cup \{e_j\})} [\hat{\gamma}(R \cup e_j) - \hat{\gamma}(R' \cup \{e, e_j\})].
\end{aligned}$$

Again from [11], $(c_j - x_j - 1) \sum_e \hat{\gamma}(R) - \hat{\gamma}(R \cup \{e\}) \geq \sum_{e_j} \sum_e \hat{\gamma}(R \cup \{e_j\}) - \hat{\gamma}(R \cup \{e_j, e\})$. Hence proved. □

Algorithm 1 provides a simple greedy algorithm. Here, to estimate $\sigma_C(\mathbf{x})$ when vaccines are uniformly at random allocated within groups, we apply the Sample Average Approximation (SAA) framework. Let $\mathcal{L} \subset \mathcal{R}(x)$, denote a sample set from the set of all possible allocations. $\sigma_C(\mathbf{x}) \approx \hat{\sigma}_C(\mathbf{x}) = \frac{1}{|\mathcal{L}|}\sum_{R\in\mathcal{L}} \gamma_C(R)$, Kempe et al. [10] show that $\gamma_C(R)$ can be estimated by sampling from the set of live-edge graphs. A live-edge graphs T is generated as follows: for each node $v \in V$, independently select at most one of its incoming edges with probability p_{uv}, and with probability $1 - \sum_{u:(u,v)\in E} p_{uv}$ no edge is selected. Let this sample set be denoted by $\mathcal{M}$. This approach takes $O(|\mathcal{M}||\mathcal{L}|(|E| + |V|))$ time to estimate $\sigma_C(\mathbf{x})$, and $O(mn|\mathcal{M}||\mathcal{L}|(|E| + |V|))$ for the full greedy algorithm, which is not practical for large networks. However, we can speed up this naive greedy algorithm.

Algorithm 2. GREEDY-LT

Require: Graph G, group set C, seed set A, and budget m
1: Merge seed set A to I
2: Sample live-edge graphs $\mathcal{M} = \{T^I_{X_1}, \ldots, T^I_{X_{|\mathcal{M}|}}\}$
3: For each T^I_X, calculate $r(u, T^I_X)$ for all nodes (in parallel)
4: Set $\mathbf{x} = \mathbf{0}$
5: **for** $j = 1$ to m **do**
6: **for** each T^I_X and C_i **do**
7: pick an edge $e^{C_i}_X$ at random for C_i and T^I_X
8: **end for**
9: $C^* = \arg\max_{C_i} \sum_{e^{C_i}_X \in T^I_X} (r(I, T^I_X) - r(I, T^I_X \setminus e^{C_i}_X))$
10: $x_{C^*} = x_{C^*} + 1$
11: **for** each T^I_X **do**
12: If $e^{C^*}_X(u, v) \in T^I_X$, remove edge $e^{C_i}_X$ and update $r(n, T^I_X)$ for node n (in parallel)
13: **end for**
14: **end for**
15: **return x**

Speed-Up of the Greedy Algorithm: GREEDY-LT. Since a live-graph sampled from $\mathcal{M}$ is a tree, we can denote it as T^s_X where s is the root, and $r(u, T^s_X) = |\{v|v \in subtree(u)\}|$, i.e., the number of nodes that are under the subtree of u in T^s_X. GREEDY-LT is summarized in Algorithm 2. It first merges all seeds into a 'supernode' s and samples $|\mathcal{M}|$ live-edge graphs, and then compute $r(u, T^s_X)$ in parallel for all nodes in all the live graphs (Lines 1-3). After that we greedily select m vaccines (Lines 4-10): we initially set the allocation vector $\mathbf{x} = \mathbf{0}$, and in each iteration, for each group C_i, we calculate the marginal loss $\Delta_{C_i,s}(\mathbf{x} + \mathbf{e_i}) = \sum_{e(u,v)\in T^s_X} r(s, T^s_X) - r(s, T^s_X \setminus e)$, i.e., we randomly pick one edge from each group for each live-edge graph, then sum their marginal losses up over T^s_X as C_i's marginal loss. Note that $r(s, T^s_X) - r(s, T^s_X \setminus e) = r(v, T^s_X) + 1$, where node v is the endpoint of e [11]. We pick the group C^* with the maximum marginal loss. Finally we removed the edge that has been picked, and update $r(u, T^s_X)$ in parallel (Lines 11-13). There are two cases to update T^s_X if $e(u, v) \in T^s_X$: (1) for v's children, we can remove them because it is not reachable from s; (2) for any ancestor a of v, $r(a, T^s_X \setminus e) = r(a, T^s_X) - r(v, T^s_X) - 1$, which can be done in constant time. Following Theorem 1, GREEDY-LT is a $(1 - 1/e - \epsilon)$-approximation algorithm where ϵ is the approximation factor for estimating $\sigma_C(\mathbf{x})$.

Running Time of GREEDY-LT. Calculating all $r(u, T^I_X)$ costs $O(|\mathcal{M}||V|)$ time since we can traverse T^I_X once to get all values of $r(u, T^I_X)$. And greedily choosing m vaccine allocation needs $O(mn|\mathcal{M}||V|)$. Hence, the serial version of GREEDY-LT costs $O(mn|\mathcal{M}||V|)$. Note that in practice, we can speed it up by computing and updating $r_i(u, T^I_X)$ in parallel. In addition, since T^I_X is tree, the increasing difference property still holds, hence we can accelerate GREEDY-LT by "lazy evaluation" [36], [40] as well.

4.2 Node Deletion under LT Model

Our algorithm for the node version of the GROUP IMMUNIZATION problem is also the greedy Algorithm 1, as in the edge version in Section 4.1. Without loss of generality, we also assume that all seed nodes in A are merged, and drop the subscript A from $\sigma'_{C,A}(\mathbf{x})$, denoting it by $\sigma'_C(\mathbf{x})$. Our analysis relies on proving that the function $f'(\mathbf{x}) = \sigma'_C(\mathbf{0}) - \sigma'_C(\mathbf{x})$ in Problem 2 satisfies the properties (P_1), (P_2) and (P_3) from Section 4.1, as discussed below.

Lemma 3. *The function* $f'(\mathbf{x}) = \sigma'_C(\mathbf{0}) - \sigma'_C(\mathbf{x})$ *satisfies the properties* (P_1), (P_2) *and* (P_3).

Proof. The proof is in the appendix. □

Lemma 3 suggests that Theorem 1 holds for node version as well: GREEDY algorithm will provide a $(1-1/e)$-approximate solution. We extend GREEDY-LT (Algorithm 2) to the node version: instead of randomly pick edges (Line 7), we randomly pick nodes to calculate the marginal loss (Line 9), and remove the corresponding nodes (Line 12). The observation is that calculating the marginal loss of removing node v in C in constant time holds here as well, i.e., $r(I, T^I_X) - r(I, T^I_X \setminus v) = r(v, T^I_X) + 1$. Hence, the updating process is the same as the edge version of GREEDY-LT.

4.3 Edge Deletion for Spectral Radius

We propose three algorithms for Problem 3 (edge immunization based on spectral radius) with different trade-offs of quality and running time: the first one, SDP, is a constant factor approximation algorithm that minimizes the actual eigendrop; the second algorithm, GROUPGREEDYWALK, is a bicriteria approximation algorithm based on hitting-walks; the third algorithm, LP, is an Linear Programming (LP) based method which uses an estimation of the eigendrop.

SDP is a constant-factor approximation algorithm, which gives us good results, but it is very slow with a $O(|V|^4 \text{polylog}(|V|))$ time complexity. Hence, we develop GROUPGREEDYWALK, a bicriteria approximation algorithm based on hitting closed walks [13]. Though GROUPGREEDYWALK loses a little quality compared to SDP, it is faster with a $O(sm^2|V|^3)$ time complexity (where s corresponds to the number of samples, described later). However, it may still not be scalable to very large networks with millions of nodes. Therefore, we come up with LP, a linear programming based heuristic whose time complexity depends only on the number of groups, not graph size. In reality, the number of groups in a group is typically much smaller than the number of nodes. Hence, LP is much faster than SDP and GROUPGREEDYWALK.

And experimental results demonstrate that it is scalable to networks with millions of nodes, and provides competitive empirical performance (see Section 5).

Note that even though SDP and GROUPGREEDYWALK may not be scalable to very large networks, both have proven performance guarantee. In addition, they are not merely of theoretical interest: they can be used as a baseline to assess the performance of faster heuristics on smaller networks.

Next, we will introduce the SDP algorithm (Section 4.3.1), the GROUPGREEDYWALK algorithm (Section 4.3.2), and the LP heuristic (Section 4.3.3) respectively.

4.3.1 SDP: A Constant Factor Approximation Algorithm

Let $G(V, E)$ be a graph whose edge set is partitioned into n groups $C_1, \ldots, C_n$. Let $\mathbf{x}$ be the edge allocation vector. For an edge (u, v), let $g(u, v)$ denote the index of the group to which (u, v) belongs. Let $G(\mathbf{x})$ be the random graph obtained by removing each edge in C_i with probability $p_i = x_i/c_i$, where $c_i = |C_i|$. Let $\mathbf{M}(\mathbf{x})$ be its adjacency matrix and $\lambda_{\mathbb{E}}(\mathbf{x}) = \mathbb{E}[\lambda(\mathbf{M}(\mathbf{x}))]$ be the expected spectral radius

$$(\mathbf{M}(\mathbf{x}))_{uv} = \begin{cases} 1, & \text{with prob. } (1 - p_{g(u,v)}) \text{ if } (u, v) \in E(G), \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

Let $\mathbf{M}_{\mathbb{E}}(\mathbf{x}) = \mathbb{E}[\mathbf{M}(\mathbf{x})]$ be the expectation of the adjacency matrix of $G(\mathbf{x})$

$$(\mathbf{M}_{\mathbb{E}}(\mathbf{x}))_{uv} = \begin{cases} 1 - p_{g(u,v)}, & \text{if } (u, v) \in E(G), \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

The problem is to find the optimal allocation, i.e., the $\mathbf{x}$ for which $\lambda_{\mathbb{E}}(\mathbf{x})$ is minimized. We will denote this value by $\lambda_{\mathbb{E}}^{\min} := \min_{\mathbf{x}} \lambda_{\mathbb{E}}(\mathbf{x})$.

Remark 4.1. In the SDP formulation, for ease of analysis, we replace the hard budget constraint by an expected budget constraint, i.e., the expected size of the vaccine allocation vector $\mathbf{x}$ is m. This is not a problem since, in reality, the budget is sufficiently high ($\gg \log n$). Hence, with high probability, the number of vaccines in the solution will be very close to the expected budget. Given this small difference, we can force the number of vaccines to be within the budget constraints, with very little effect on the performance.

The SDP *Formulation: Finding the Allocation* $\mathbf{x}$ *with Minimum* $\lambda(\mathbf{M}_{\mathbb{E}}(\mathbf{x}))$. Note that, $\mathbf{M}_{\mathbb{E}}(\mathbf{x})_{uv} = (1 - p_{g(u,v)})$, if $(u, v) \in E(G)$. We use a simple SDP to find the allocation which minimizes $\lambda(\mathbf{M}_{\mathbb{E}}(\mathbf{x}))$ and meets the budget constraint m

$$\begin{array}{ll} \text{minimize} & \mathsf{t} \\ \text{subject to} & 0 \le p_i \le 1, \text{ for } i = 1, \ldots, n \\ & \sum_i p_i |C_i| \le m, \\ & tI - \mathbf{M}_{\mathbb{E}}(\mathbf{x}) \succeq 0. \end{array} \tag{4}$$

Let $\mathbf{x}_{\min}$ denote the allocation vector corresponding to the solution of the SDP.

Analysis: Relating $\lambda_{\mathbb{E}}^{\min}$ *to* $\lambda(\mathbf{M}_{\mathbb{E}}(\mathbf{x}_{\min}))$. One can use the following result by Lu and Peng [41] to bound $\lambda_{\mathbb{E}}(\mathbf{x})$ with respect to $\lambda(\mathbf{M}_{\mathbb{E}}(\mathbf{x}))$.

Theorem 2 ([41]). *Consider an edge-independent random graph* H. *Let* $\mathbf{M}(H)$ *denote its adjacency matrix and* $\mathbf{M}_{\mathbb{E}}(H) = \mathbb{E}[\mathbf{M}(H)]$. $\Delta_{\mathbb{E}}(H)$ *denotes the maximum expected degree. If* $\Delta_{\mathbb{E}}(H) \gg \log^4 |V|$, *then, almost surely* $\lambda_i(\mathbf{M}(H)) - \lambda_i(\mathbf{M}_{\mathbb{E}}(H))| \le (2 + o(1))\sqrt{\Delta_{\mathbb{E}}(H)}$, *for* $i = 1, \ldots, |V|$.

Recall that $\mathbf{x}_{\min}$ is the output of SDP (4), and it corresponds to the allocation vector which minimizes $\lambda(\mathbf{M}_{\mathbb{E}}(\mathbf{x}))$ over all $\mathbf{x}$. Let $\Delta_{\mathbb{E}}(\mathbf{x}_{\min})$ denote the maximum expected degree of $G(\mathbf{x}_{\min})$. The following lemma proves that the SDP formulation gives us an approximation algorithm with constant factor $O(\sqrt{\Delta_{\mathbb{E}}(\mathbf{x}_{\min})})$.

Lemma 4. *If* $\mathbf{x}_{\min}$ *is such that* $\Delta_{\mathbb{E}}(\mathbf{x}_{\min}) \gg \log^4 |V|$, *then,* $\lambda_{\mathbb{E}}^{\min} \le \lambda(\mathbf{M}_{\mathbb{E}}(\mathbf{x}_{\min})) + (2 + o(1))\sqrt{\Delta_{\mathbb{E}}(\mathbf{x}_{\min})} + 1$.

Proof. Let $z = \lambda(\mathbf{M}_{\mathbb{E}}(\mathbf{x}_{\min})) + (2 + o(1))\sqrt{\Delta_{\mathbb{E}}(\mathbf{x}_{\min})}$. Applying Theorem 2 to $G(\mathbf{x}_{\min})$, $\lambda(\mathbf{M}(\mathbf{x}_{\min})) \le z$ almost surely. In fact, for $\Delta_{\mathbb{E}}(\mathbf{x}_{\min}) \gg \log^4 |V|$, it can be shown that $\Pr\big(\lambda(\mathbf{M}(\mathbf{x}_{\min})) \ge z\big) \le 1/|V|$ (see [41, proof of Theorem 6]). Noting that $\lambda(\mathbf{M}(\mathbf{x}_{\min})) \le \lambda(\mathbf{M})$,

$$\begin{aligned} \lambda_{\mathbb{E}}(\mathbf{x}_{\min}) &= \mathbb{E}[\lambda(\mathbf{M}(\mathbf{x}_{\min}))] \\ &\le \Pr\big(\lambda(\mathbf{M}(\mathbf{x}_{\min})) \le z\big) \cdot z \\ &\quad + \Pr\big(\lambda(\mathbf{M}(\mathbf{x}_{\min})) \ge z\big) \cdot \lambda(\mathbf{M}) \\ &\le 1 \cdot z + \Big(\frac{1}{|V|}\Big) \cdot \lambda(\mathbf{M}) \; < \; z + 1\,. \end{aligned}$$

By definition, $\lambda_{\mathbb{E}}^{\min} \le \lambda_{\mathbb{E}}(\mathbf{x}_{\min})$. Therefore, $\lambda_{\mathbb{E}}^{\min} \le \lambda_{\mathbb{E}}(\mathbf{x}_{\min}) \le z + 1$. Hence, proved. □

Running Time. The SDP step (Eq. (4)) dominates the running time of this algorithm, which is $O(|V|^4 \text{polylog}(|V|))$.

4.3.2 GROUPGREEDYWALK: A Bicriteria Approximation Algorithm

As shown above, SDP with a $(|V|^4 \text{polylog}(|V|))$ time complexity, is too slow for large networks. In this section, we leverage the technique of hitting closed walks [13] for the GROUP IMMUNIZATION problem, and propose a bicriteria approximation algorithm called GROUPGREEDYWALK.

Saha et al. [13] studied the problem of minimizing the spectral radius under a given threshold by removing the smallest number of edges, and developed a greedy based approximation algorithm for it. Different from their work, our goal is to distribute a given budget of vaccines to groups to minimize the spectral radius as small as possible. We can adapt their greedy algorithm to the group immunization, by choosing groups with maximum marginal gain of hitting closed walks. However, it is not clear whether this works, as we need to consider all "possible worlds" for group immunization.

In graph G, a closed walk is a sequence of nodes starting and ending at the same node, with two consecutive nodes adjacent to each other. Closed k-walk is a walk with length k. Let walks (e, G, k) denote the number of closed k-walks in G containing $e = (i, j)$. We say that an edge set E hits a walk w if w contains an edge from E. Recall that $G(\mathbf{x})$ is a random graph obtained by removing a random subset of x_i edges in C_i, where $C_1, \ldots, C_n$ is a partition of the edge set E. Let $\mathcal{W}(G, k)$ be the set of all walks of length k in the graph G. Let $n_k(G, e)$ denote the number of walks of length k in G that pass through edge e. Similarly, let $n_k(G, S)$ denote the walks of length k in G that pass though edges in the set S. Let $n_k(G) = n_k(G, E) = |\mathcal{W}(G, k)|$ denote

the number of walks with length k in G. Here we focus on walks of a fixed length $k = \theta(\log |V|)$. Note that for $G(\mathbf{x})$, $n_k(G(\mathbf{x}))$ is a random variable.

Algorithm 3. GROUPGREEDYWALK (G, m)

Require: Graph G, group set C, and budget m
1: $\mathbf{x} = 0$
2: **for** $j = 1$ to m **do**
3: $i = \arg\max_{k=1,\ldots,n}$ EXPCOUNTWALKS$(\mathbf{x} + \mathbf{e}_k)-$ EXPCOUNTWALKS$(\mathbf{x})$
4: $\mathbf{x} = \mathbf{x} + \mathbf{e}_i$
5: **end for**
6: **return x**

Algorithm 3 gives the pseudocode of our GROUPGREEDYWALK algorithm. EXPCOUNTWALKS$(G, \mathbf{x})$ returns the expected number of walks surviving in $G(\mathbf{x})$. Note that EXPCOUNTWALKS is different from COUNTWALKS in [13], as it returns the expected number of hitting walks when a vaccine allocation vector $\mathbf{x}$ is assigned to groups, while COUNTWALKS in [13] is based on removing a set of edges given a budget constraint. The idea of GROUPGREEDYWALK is that, each time we select a group C_i with the maximum marginal gain in EXPCOUNTWALKS$(G, \mathbf{x})$, when allocating one vaccine to C_i.

Algorithm 3 follows the framework of the individual based GREEDYWALK algorithm [13]. Instead of picking edges, it chooses *groups* to maximize marginal gain of eigendrop. The main challenge here is to show GROUPGREEDYWALK is a provable approximation algorithm. Let $\mathbf{x}^{opt}(m)$ be the optimum solution corresponding to budget m , and $T = \lambda_1(G(\mathbf{x}^{opt}(m)))$ (the spectral radius after vaccine allocation for the optimum solution). We can prove the following theorem:

Theorem 3. *Let $\mathbf{x}^{opt}(m)$ be the optimum solution corresponding to budget m of edges removed. Let $\mathbf{x}^g$ be the allocation returned by GROUPGREEDYWALK $(G, c_1 m \log^2|V|)$, for a constant c_1. Then we have $\lambda_1(G(\mathbf{x}^g)) \leq c'T$ for a constant c', where $\lambda_1(G(\mathbf{x}^g))$ is the spectral radius after allocating vaccines based on $\mathbf{x}^g$.*

Remark 4.2. Theorem 3 shows that GROUPGREEDYWALK is a $(c_1 \log^2|V|, c')$-bicriteria approximation algorithm. Different from the analysis of traditional approximation algorithms, in order to bound the result of GROUPGREEDYWALK w.r.t to the optimal solution, we need a larger budget $c_1 \log^2|V|m$. Typically, $\log^2|V|$ is much smaller than the budget m. And when the budget m is very large, the marginal gain of eigendrop for a larger budget $c_1 \log^2|V|m$ will tend to be very close to the marginal gain of eigendrop for the budget m. Hence, adding such small factor into the budget m will have little effect on the performance.

We will use Lemmas 5 and 6 to prove this theorem. Intuitively, Lemma 5 shows the expected spectral radius is upperbounded by T if the number of walk $k = O(\log |V|)$; while Lemma 6 shows that the expected number of walks with length k can be upperbounded by T as well.

Lemma 5. *If $E[n_k(G(\mathbf{x})] = O(|V|2^k T^k)$ for $k = O(\log |V|)$, then $E[\lambda_1(G(\mathbf{x}))] \leq c_3 T$ for a constant c_3.*

Proof. The proof is in the appendix. □

Lemma 6. *Let $\mathbf{x}^{opt}(m)$ be the optimum allocation such that $T = E[\lambda_1(G(\mathbf{x}^{opt}(m)))]$. Let $\mathbf{y}$ be defined as*

$$y_i = \begin{cases} x_i^{opt}, & \text{if } x_i^{opt} \leq m_i/2, \\ m_i & \text{otherwise,} \end{cases}$$

where m_i is the number of edges in group C_i. Then, we have $E[n_k(G(\mathbf{y}))] \leq |V|2^k T^k$.

Proof. The proof is in the appendix. □

Now, we prove Theorem 3.

Proof of Theorem 3. Let $g(\mathbf{x})$ denote the expected number of walks in $\mathcal{W}(G, k)$ hit by the edges that are removed in $G(\mathbf{x})$. Then $g(\mathbf{x})$ has the diminishing returns property, i.e., for $\mathbf{x} \leq \mathbf{x}'$, we have $g(\mathbf{x} + \mathbf{e}_i) - g(\mathbf{x}) \geq g(\mathbf{x}' + \mathbf{e}_i) - g(\mathbf{x}')$. The proof of the diminishing returns follows the proof of Lemma 2.

We will compare $g(\mathbf{x}^g)$ to $g(\mathbf{y})$ where $\mathbf{y}$ is defined as

$$y_i = \begin{cases} x_i^{opt}, & \text{if } x_i^{opt} \leq m_i/2, \\ m_i & \text{otherwise,} \end{cases}$$

where m_i is the number of edges in group C_i. Note that $\sum_i y_i \leq 2\sum_i x_i^{opt} \leq 2m$.

Let $\mathbf{x}^{(i)}$ denote the vector after ith iteration of GROUPGREEDYWALK. Since $g(\mathbf{x})$ has the diminishing returns property, it follows the proof of Theorem 1 that $f(\mathbf{x}^{(i)}) \geq (1 - (1 - \frac{1}{2m})^i) f(\mathbf{y})$. Therefore, for $i = O(m \log^2|V|)$, we have $1 - (1 - \frac{1}{2m})^{O(m \log^2|V|)} \geq 1 - (1/e)^{\log^2|V|} \geq 1 - \frac{1}{|V|^{\log|V|}} \geq 1 - \frac{1}{N}$, where N is the number of total walks in the original graph G.

From Lemma 6, we have $E[n_k(G(\mathbf{y}))] \leq |V|^c T^k$ for a constant c. This implies $f(\mathbf{y}) \geq N - |V|^c T^k$. Therefore, $f(\mathbf{x}^g) \geq (1 - \frac{1}{|V|})(N - |V|^c T^k) \geq N - 1 - |V|^c T^k$. This implies that $n_k(G(\mathbf{x}^g)) \leq O(|V|^c T^k)$. From Lemma 5, it follows that $\lambda_1(G(\mathbf{x}^g)) \leq c'T$. □

Implementation Notes. Given the adjacency matrix A of G, the number of k-length walks from u to v is given by A^{k-1}_{uv}. It also corresponds to the number of walks hit by the edge (u, v). We implement the algorithm as follows. In each iteration, we randomly sample a set of edges of the G according to $\mathbf{x}$. For each sample, we compute the expected decrease in the number of walks for the removal of one edge in group i (for computing the effect of allocation vector $\mathbf{x} + \mathbf{e}_i$) as follows: We construct $G(\mathbf{x})$, compute $A' = A(G(\mathbf{x}))^{k-1}$ and take the average over all $A'(u, v)$ elements where (u, v) belongs to group i. We perform this for each sample (number of samples is s) and take the average over all the samples. Finally, we choose that i which gives the maximum average and update $\mathbf{x}$ by adding $\mathbf{e}_i$ to it.

Running Time. For budget m, A^{m-1} can be computed in time $O(m^2|V|^3)$. For each sample of $\mathbf{x}$, we compute $A(G(\mathbf{x}))^{m-1}$. Note that, computing the effect of removing $\mathbf{e}_i$ for each sample takes only $O(|V|^2)$ time. Therefore, for a sample size of s, the algorithm overall takes $O(sm^2|V|^3)$ time. If $m = O(\log |V|)$, the time complexity is $O(s|V|^3 \log^2|V|)$.

4.3.3 LP: A Fast Heuristic

GroupGreedyWalk is a good approximation algorithm like SDP, however, it may not be scalable to very large networks with millions of nodes. In this section, the propose a much faster heuristic based on estimating eigendrop.

The eigendrop when removing edges in the set E_T can be approximated by $\phi(T) = \sum_{(i,j)\in E_T} \mathbf{M}_{ij} u_i u_j$ where $\mathbf{Mu} = \lambda \mathbf{u}$ and $\mathbf{u} = (u_1, \ldots, u_i, \ldots)$ [29]. Given the allocation vector $\mathbf{x}$, the expected drop in spectral radius is then given by

$$\begin{aligned}\mathbb{E}[\Delta\lambda] &\approx \phi(\mathbf{x}) \\ &= \sum_{i,j\in E} \mathbf{M}_{ij} u_i u_j \Pr\big((i,j) \text{ is removed}\big) \\ &= \sum_{a\in C} \sum_{(i,j)\in C_k} \mathbf{M}_{ij} u_i u_j x_a \,.\end{aligned} \tag{5}$$

If we define $\alpha_a = \sum_{(i,j)\in C_a} \mathbf{M}_{ij} u_i u_j$, then, $\phi(\mathbf{x}) = \sum_a \alpha_a x_a$. We want to maximize $\phi(\mathbf{x})$ subject to the budget constraints. This can be formulated as a linear program as given below

$$\begin{array}{ll}\text{maximize} & \sum_a \alpha_a x_a \\ \text{subject to} & \sum_a x_a |C_a| \le m \\ & 0 \le x_a \le 1.\end{array} \tag{6}$$

Running Time. The LP takes $O(n^4)$ time where n is the number of groups. Note that it is not a function of the graph size. Typically, the number of groups is small, hence this algorithm is very fast.

4.4 Node Deletion for Spectral Radius

Here, we propose an algorithm for solving Problem 4: the group node immunization problem with respect to eigendrop. It is based on the approximate eigendrop method which was discussed in Section 4.3. The eigendrop when removing nodes in S can be approximated as follows [7]:

$$\Delta\lambda \approx \phi(S) = \sum_{j\in S} 2\lambda u_j^2 - \sum_{i,j\in S} \mathbf{M}_{ij} u_i u_j, \tag{7}$$

where $\mathbf{Mu} = \lambda \mathbf{u}$ and $\mathbf{u} = (u_1, \ldots, u_i, \ldots)$. Recall that C is the set of groups and $\mathbf{x} = (x_1, \ldots, x_i, \ldots)$ is the allocation vector where, x_i is the fraction of nodes vaccinated in group C_i. For the group vaccination problem, the expected eigendrop can be approximated by applying (7) as follows:

$$\begin{aligned}\mathbb{E}[\Delta\lambda] \approx \phi(\mathbf{x}) = &\sum_{j\in V} 2\lambda u_j^2 \Pr(j \text{ is vaccinated}) \\ &- \sum_{i,j\in V} \mathbf{M}_{ij} u_i u_j \Pr(i \;\&\; j \text{ are vaccinated}) \,.\end{aligned} \tag{8}$$

Let $g(v)$ denote the index of the group to which v belongs to, i.e., if $v \in C_i$, then, $g(v) = i$. The probability that j is vaccinated is $x_{g(j)}$ and the probability that both i and j are vaccinated is

$$\Pr(i \;\&\; j \text{ are vaccinated}) = \begin{cases} x_{g(i)} x_{g(j)}, & \text{if } g(i) \ne g(j), \\ x_{g(i)}^2 \frac{|C_{g(i)}|}{|C_{g(i)}|-1}, & \text{otherwise.}\end{cases} \tag{9}$$

Applying the above to (8),

$$\begin{aligned}\phi(\mathbf{x}) = &\sum_a \sum_{j\in C_a} 2\lambda u_j^2 x_a - \sum_a \sum_{i,j\in C_a} \mathbf{M}_{ij} u_i u_j x_a^2 \frac{|C_a|}{|C_a|-1} \\ &- \sum_{a\ne b} \sum_{i\in C_a, b\in C_b} \mathbf{M}_{ij} u_i u_j x_a x_b \,.\end{aligned}$$

Observing that $\mathbf{M}_{ij}$, u_i and x_a are constants, defining $\alpha_a = \sum_{j\in C_a} 2\lambda u_j^2 \frac{|C_a|}{|C_a|-1}$, $\beta_a = \sum_{i,j\in C_a} \mathbf{M}_{ij} u_i u_j$, and $\Gamma_{ab} = \sum_{i\in C_a, j\in C_b} \mathbf{M}_{ij} u_i u_j$, we get,

$$\phi(\mathbf{x}) = \sum_a \alpha_a x_a - \sum_a \beta_a x_a^2 - \sum_{a\ne b} \Gamma_{ab} x_a x_b \,.$$

Our aim is to find that $\mathbf{x}$ which maximizes $\phi(\mathbf{x})$. This can be formulated as a quadratic program

$$\begin{array}{ll}\text{minimize} & \sum_a \beta_a x_a^2 + \sum_{a\ne b} \Gamma_{ab} x_a x_b - \sum_a \alpha_a x_a \\ & = \frac{1}{2}\mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{c}^T \mathbf{x} \\ \text{subject to} & \sum_a x_a |C_a| \le B \\ & 0 \le x_a \le 1,\end{array} \tag{10}$$

where, $\mathbf{Q}_{aa} = 2\beta_a$ and for $a \ne b$, $\mathbf{Q}_{ab} = 2\Gamma_{ab}$ and $\mathbf{c}_a = -\alpha_a$. If $\mathbf{Q}$ is not semi-definite, the problem is NP-Hard [42]. In that case, we use a low-rank matrix $\widehat{\mathbf{Q}}$ formed by all its eigenvectors corresponding to non-negative eigenvalues. The QP on $\widehat{\mathbf{Q}}$ can be solved in polynomial time using the ellipsoid method [42]. Let $\widehat{\phi}(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T \widehat{\mathbf{Q}} \mathbf{x} + \mathbf{c}^T \mathbf{x}$, and $\mathbf{x}_{\mathbf{Q}}$, $\mathbf{x}_{\widehat{\mathbf{Q}}}$ correspond to the best allocation vectors corresponding to $\mathbf{Q}$ and $\widehat{\mathbf{Q}}$ respectively. The next lemma shows that $\mathbf{x}_{\widehat{\mathbf{Q}}}$ is a good approximation to $\mathbf{x}_{\mathbf{Q}}$.

Lemma 7. *$|\widehat{\phi}(\mathbf{x}_{\widehat{\mathbf{Q}}}) - \phi(\mathbf{x}_{\mathbf{Q}})| \le \frac{n}{2} \cdot \|\mathbf{Q} - \widehat{\mathbf{Q}}\|_F$, where n is the number of groups in the graph.*

Proof. The proof is in the appendix. □

Running Time. The QP takes $O(n^4)$ time. Again, note that n is the number of groups. Hence, it is fast when the number of groups is small.

5 Empirical Study

We present a detailed experimental evaluation now.

5.1 Experimental Setup

We implemented the algorithms in Python,[2] and conducted the experiments using a 4 Xeon E7-4850 CPU with 512 GB of 1,066 Mhz main memory.

Datasets. Table 2 briefly summarizes the dataset. We run our experiments on multiple datasets, which were chosen for their size as well as different domains where the Group Immunization problem is especially applicable. Note that all our datasets are networks, not diffusion traces. If diffusion traces are provided as inputs instead of a network, there are state-of-the-art algorithms (such as [43]) which can be applied to learn edge weights first, and then apply our algorithm.

2. Code: http://people.cs.vt.edu/yaozhang/group-immu/

TABLE 2
Datasets

Dataset	Num. of nodes	Num. of edges	Num. of groups
SBM	1,500	5,000	20
Protein	2,361	7,182	13
OregonAS	10,670	22,002	100
YouTube	50 K	450 K	5,000
Portland	0.5 million	1.6 million	91
Miami	0.6 million	2.1 million	91

1) SBM (Stochastic Block Model) [44] is a well-known model to generate synthetic graphs with groups. We generate small networks from the Stochastic Block Model to test the effectiveness of all our methods.
2) Protein[3] is a protein-protein interaction network in budding yeast. There are 13 classes of proteins, which are naturally treated as groups. It is a biological network, where our immunization algorithms can be potentially applied to block protein interactions.
3) OregonAS[4] is the Oregon AS router graph collected from the Oregon router views, and groups here are based on router conductivities. We use Louvain [45], a fast community detection algorithm to specify groups. It is a computer network where our algorithms can be used to stop malware outbreaks.
4) YouTube[5] is a friendship network in which users can form groups. We create an induced graph by selecting nodes that are in the top 5,000 communities. It is a social media network where we can apply our algorithms to control rumor spread.
5) Portland and Miami are social-contact graphs based on detailed microscopic simulations of large US cities, which has been used in national smallpox and influenza modeling studies using the SIR model [2]. We divided people into groups by ages ranging from 0-90 (hence 91 groups in both networks). They are both contact networks where our algorithms can be adopted to minimize virus propagation.

Settings. For LT model, we uniformly randomly choose 1 percent nodes as the infected nodes (seeds) at the start. And we use the same method in [11] to generate the probabilities on the edges: for a node v, we assign each its incoming edge (u, v) with a probability $\hat{p}_{uv}$ uniformly at random, then we uniformly randomly give a probability w_v to v representing v's incoming edges fail to activate it. Then we get the normalized weight $p_{uv} = \hat{p}_{uv}/(\sum_{u \in V} \hat{p}_{uv} + w_v)$. We construct 1,000 live-edge graphs in our algorithm for LT model. For robustness, each data point we show is the mean of 1,000 runs of randomly sampling removed edges/nodes from groups. In the edge deletion version, edge communities are induced from node communities, i.e., for an edge $e = (u, v)$, if both u and v belong to a group C_t, then $e \in C_t$, otherwise it belongs to group $C_{ij} = \{e_{uv} | u \in C_i, v \in C_j\}$.

Baselines. As we are not aware of any direct competitor tackling our group immunization problems, we construct three baselines for both node and edge deletion to better judge their performance. Analogous versions of these baselines have been regularly used in state-of-the-art individual immunization studies [7], [13], [29].

(1) RANDOM: uniformly randomly assign vaccines to groups for both node deletion and edge deletion.
(2) DEGREE: for node deletion, we calculate the average degree d_{C_i} of each group C_i, and independently assign vaccines to C_i with probability $d_{C_i}/\sum_{C_k \in C} d_{C_k}$; for edge deletion, we first calculate the product degree d_e [30] of each edge $e = (u, v)$, i.e., $d_e = d_u * d_v$, then similar to node deletion, we calculate the average product degree d_{C_i} of C_i, and assign vaccines to C_i with probability $d_{C_i}/\sum_{C_k \in C} d_{C_k}$.
(3) EIGEN: Eigenvalue centrality has been widely used in the immunization literature [7], [29], even as a baseline for LT model [11]. Let $\mathbf{u}$ be the eigenvector corresponding to the first eigenvalue of the graph. The eigenscore of node a is u_a, while the eigenscore of edge $e(a, b)$ is $|u_a u_b|$ [29]. For both node and edge deletion, we calculate the average eigenscore u_{C_i} of each group C_i, and independently assign vaccines to C_i with probability $u_{C_i}/\sum_{C_k \in C} u_{C_k}$.

Remark 5.1. Note that we do not compare and run the individual based immunization methods [7], [11] "as-is" on the original graph because these methods directly pick nodes which we do not allow in our problems. Instead, we aim to pick the best groups, and then uniformly at random allocate vaccines within the group. In addition, we did study the effect of our algorithm w.r.t. the size of groups (see Fig. 6). If each node is a group, GROUP IMMUNIZATION reduces to the individual based immunization. Indeed the reason we formulate the group immunization problems in this paper is that it is typically not feasible to force targeted individuals to be vaccinated in practice (as discussed before in the introduction).

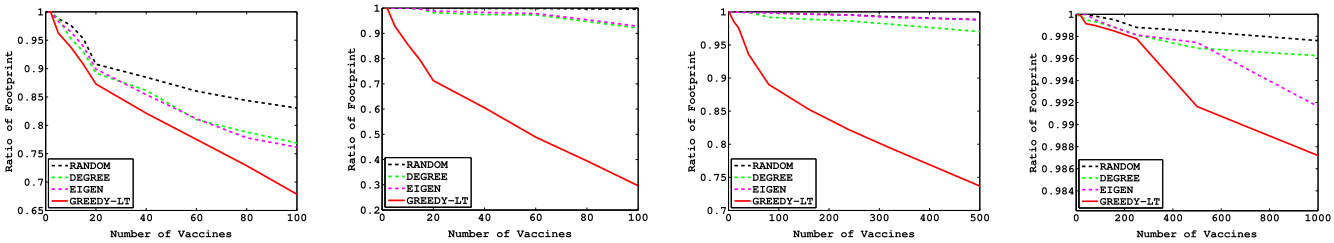
5.2 Results

In short, we demonstrate that our methods outperform other baselines on all datasets. We also show how the behaviors of our methods change as groups vary. Finally, we conduct a case study to analyze the vaccine allocations at group scale.

5.2.1 Performance

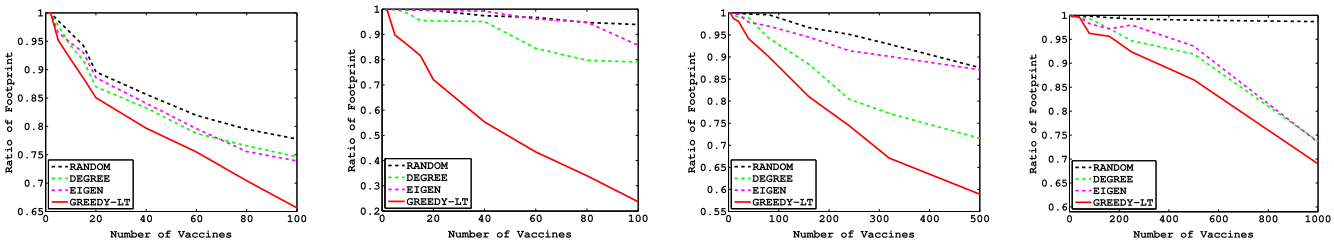
Fig. 1 shows experimental results under LT model for group edge deletion, while Fig. 2 demonstrates the results for node deletion. In all networks, GREEDY-LT consistently outperform other competitors. Since we have same budgets for both edge and node deletion, clearly node removal should perform better than edge deletion as node deletion removes more edges. Our results demonstrate this fact. As shown in Fig. 1, GREEDY-LT performs pretty well for edge deletion compared with other competitors, e.g., in YouTube, GREEDY-LT can reduce about 25 percent of the infection if 500 edges are removed, while for RANDOM, DEGREE and EIGEN, the infection almost remains the same even removing 500 edges. For node deletion (Fig. 2), GREEDY-LT performs even better: it reduces more than 30 percent of the infection given the maximum budgets.

3. http://vlado.fmf.uni-lj.si/pub/networks/data/bio/Yeast/Yeast.htm
4. http://snap.stanford.edu/data/oregon1.html
5. http://snap.stanford.edu/data/com-Youtube.html



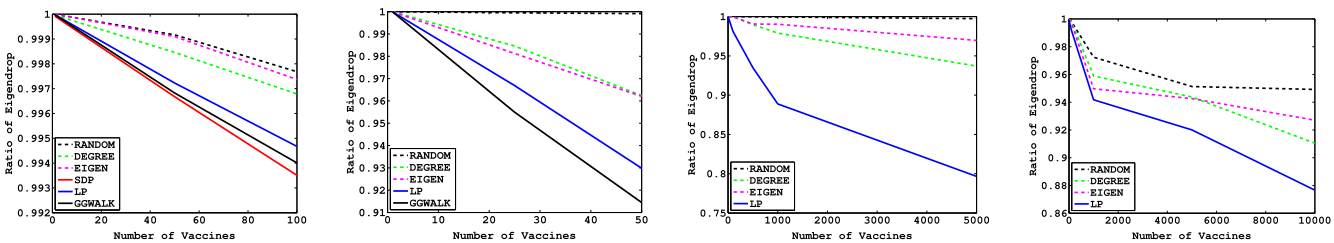
(a) SBM (b) OregonAS (c) YouTube (d) Portland

Fig. 1. Effectiveness for LT model various Real Datasets (edge deletion). Footprint ratio ($\frac{footprint\ when\ vaccines\ are\ given}{footprint\ without\ giving\ vaccines}$) versus number of vaccines. Lower is better. GREEDY-LT consistently outperforms other baseline algorithms.



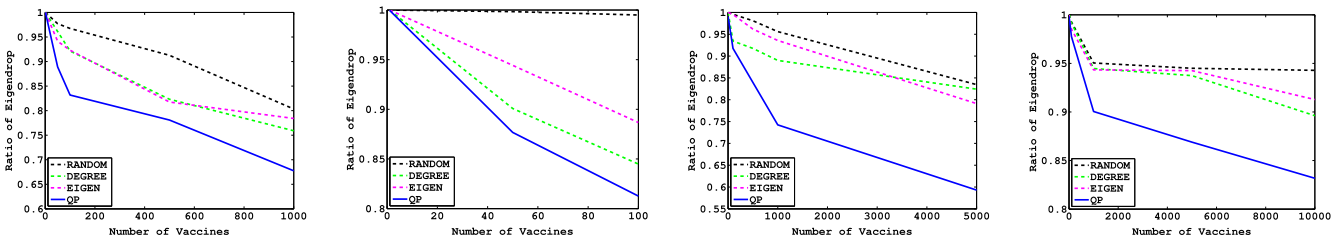
(a) SBM (b) OregonAS (c) YouTube (d) Portland

Fig. 2. Effectiveness for LT model various Real Datasets (node deletion). Footprint ratio ($\frac{footprint\ when\ vaccines\ are\ given}{footprint\ without\ giving\ vaccines}$) versus number of vaccines. Lower is better. GREEDY-LT consistently outperforms other baseline algorithms.



(a) Protein (b) OregonAS (c) YouTube (d) Portland

Fig. 3. Effectiveness for the change of the first eigenvalue various Real Datasets (edge deletion). Eigendrop ratio ($\frac{\lambda'_G}{\lambda_G}$) versus number of vaccines (λ'_G is the expected eigenvalue after allocating vaccines). Lower is better. SDP, GROUPGREEDYWALK, and LP consistently outperform other baseline algorithms.

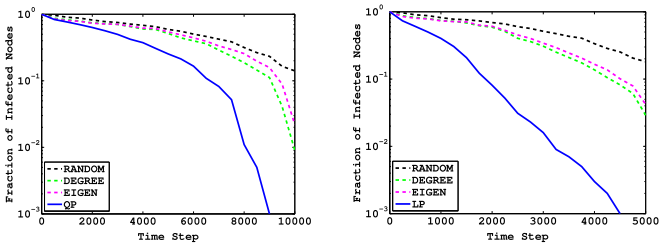


(a) Protein (b) OregonAS (c) YouTube (d) Portland

Fig. 4. Effectiveness for the change of the first eigenvalue various Real Datasets (node deletion). Eigendrop ratio ($\frac{\lambda'_G}{\lambda_G}$) versus number of vaccines (λ'_G is the expected eigenvalue after allocating vaccines). Lower is better. QP consistently outperforms other baseline algorithms.

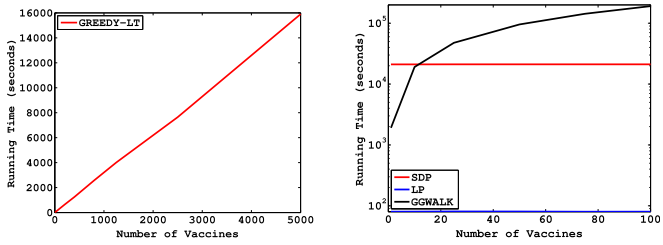
Fig. 3 shows experimental results of edge version of group immunization for spectral radius, while Fig. 4 demonstrates the results for node deletion. In all networks, SDP, GROUPGREEDYWALK, LP and QP consistently outperform other competitors. SDP gives the best results for Protein, however, it is not scalable to large networks with more than thousands of nodes. GROUPGREEDYWALK gives the second best performance, and it works for graphs with about 10 K nodes. For very large networks like YouTube and Portland with millions of nodes, approximate algorithms like SDP and GROUPGREEDYWALK can not finish within an allocated time. LP for edge deletion and QP for node deletion, perform very well for large networks. For edge deletion (Fig. 3), RANDOM, DEGREE and EIGEN cannot decrease more than 10 percent of the first eigenvalue in YouTube when 5 k vaccines are given to groups, while LP can reduce more than 20 percent of the eigenvalue. For node deletion (Fig. 4), QP can get more than twice reduction of eigenvalue compared to other competitors. When comparing between node and edge deletion, we get the same result as Figs. 1 and 2: given same vaccines to both edge and node, node removal can get a larger decrease of the spectral radius.

As mentioned above, the problems of minimizing the spectral radius are motivated by the epidemic threshold [12]; an epidemic will quickly die out if the spectral radius is very



(a) Portland (Node, Budget=10,000) (b) YouTube (Edge, Budget=5000)

Fig. 5. SIS simulations after vaccine allocation. The fraction of infected nodes (in log-scale) versus the time step. Lower is better. SDP, GROUPGREEDYWALK, and LP consistently outperform other baseline algorithms.



(a) YouTube LT model (b) SBM Spectral Radius

Fig. 7. Running Time (seconds). Running time versus number of vaccines.

small. Hence as an example, we also run the SIS model to show how effective our algorithms are to prevent an epidemic from breaking out. We assume all nodes are in the infectious states at the beginning, and the recovery rate is 0.6. Fig. 5 shows the results on Portland and YouTube for node deletion and edge deletion respectively, which is averaged over 1,000 runs (note that we got the similar results on other networks). We observe that LP and QP consistently outperform other competitors: they both have the least number of infected nodes in the network when vaccines are allocated.

5.2.2 Varying Groups

We would like to see the effect of the change of granularity of vaccine allocation. We changed the number of groups on Portland, YouTube and OregonAS. For Portland, age ranges from 0 to 90, hence there are initially 91 groups. We decrease the number of groups by randomly merging two adjacency age groups. For OregonAS, we use community detection algorithm Louvain [45] to find different number of groups. For YouTube, we randomly merge ground true communities to form smaller size of groups.

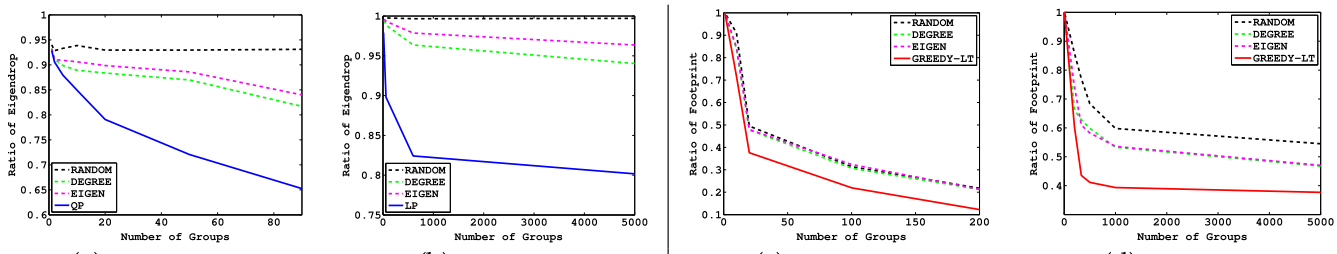
Figs. 6a and 6b show the performance of QP and LP as the number of groups changes. First, both of them outperform other baselines for Portland and YouTube. Second, as the number of groups increases, the spectral radius decreases more for all algorithms (except for RANDOM) due to the fact that the randomization of allocating vaccines deceases. The extreme case is that when there is only one group, QP, DEGREE and EIGEN are uniformly randomly allocate vaccine to the whole graph, which is exactly the same as RANDOM. On the contrary, when the number of groups is equal to the number of nodes, group immunization becomes individual immunization which is effective but much more expensive. Figs. 6c and 6d show the performance of GREEDY-LT as the number of groups varies. Similar to QP and LP, it consistently outperforms other baselines. And the performance improvement is even more obvious: when the graph size increases from 1 to 200, GREEDY-LT almost reduces 90 percent of the infection.

5.2.3 Scalability

Although our algorithms are polynomial-time, we show some running time results to demonstrate the scalability of our algorithms. Fig. 7 shows the running time of our algorithms w.r.t. the number of vaccines. We did not show the running time of RANDOM, DEGREE and EIGEN, because they are faster heuristics. First, as expected from the time complexity of GREEDY-LT, when the number of vaccines m increases, the running time of GREEDY-LT increases linearly (Fig. 7a). Second, since the time complexities of SDP and LP are irrelevant to m, as shown in Fig. 7b, the running time of them remains almost constant. Furthermore, we observe that when m is small, GROUPGREEDYWALK ran faster than SDP. As the performances of GROUPGREEDYWALK and SDP are very close, in large graphs with a relatively small budget, we could get very good solution from GROUPGREEDYWALK.

5.2.4 Case Study

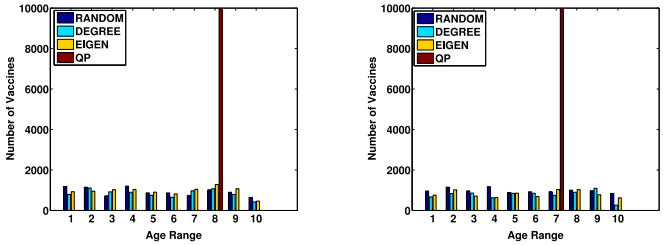
We now study the group vaccination problem on realistic social contact networks, Portland and Miami, using age based groups; as discussed earlier, age based directives are commonly used by public health agencies. Fig. 8 shows the number of vaccines assigned to different age groups, for a total of 10,000 vaccines, using the QP algorithm. We find the groups with age 70-79 and 60-66 get the maximum allocation,



(a) Portland (Node, Budget=20,000) (b) YouTube (Edge, Budget=5000) Spectral Radius

(c) OregonAS (Edge, Budget=1000) (d) YouTube (Node, Budget=1000) LT model

Fig. 6. (a) and (b): Eigendrop ratio versus number of groups. (c) and (d): Footprint ratio versus number of groups. Lower is better. Our algorithms consistently outperform other baseline algorithms as the number of groups changes as well as the size of groups changes.



(a) Portland (Node) (b) Miami (Edge)

Fig. 8. Vaccine Distributions for Portland and Miami (Budget = 10,000). Number of vaccines versus Age. (Age range '0-9': 1; '10-19': 2; '20-29': 3; '30-39': 4; '40-49': 5; '50-59': 6; '60-69': 7; '70-79': 8; '80-89': 9; '90-': 10.)

for the Portland and Miami networks, respectively. This contrasts with CDC recommendations, and the strategy proposed by Medlock et al. [1], as CDC recommendations include children through age 18, and Medlock et al. suggests to prioritization of schoolchildren and adults aged 30 to 39 years. This might be because these results do not use the detailed network structure. We believe this is an interesting result which merits further study.

6 Discussion and Conclusion

This paper addresses the problems of controlling epidemics by means of interventions that can be implemented at a group level. We formulate the Group Immunization problem in the LT model as well as SIS/SIR models (considering the spectral radius minimization) for both edge-level and node-level interventions. We develop algorithms with rigorous performance guarantees and good empirical performance for all these problem classes. Our algorithms require a diverse class of techniques, including submodular function maximization, linear programming, quadratic programming, semidefinite programming, and hitting closed walks. Finally, we evaluate them on real networks of diverse scales. We demonstrate that our algorithms significantly outperform other heuristics, and adapt to the group structure. Some of our algorithms, e.g., SDP is fairly time intensive, though it runs in polynomial time. However, it is important to keep in mind that these algorithms are expected to be run before an epidemic outbreak, where the solution quality is much more critical than the run time.

Currently our SDP and GroupGreedyWalk algorithm work for edge deletion. Developing provable approximation algorithms for node deletion by leveraging SDP and GroupGreedyWalk, can be another future direction. In addition, our formulations capture the uncertainty, lack of control and compliance at a fine granularity in immunization interventions in public health and social media. Another important practical consideration is the economies of scale that arise in such group level formulations—these could be the result of decreasing per unit cost of production or distribution within a group. Such constraints can be modeled as $\sum_{i=1}^{n} \phi_i(x_i) \leq B$, where $\phi_i(x_i)$ is a concave function and x_i is the allocation to group C_i, and B is a budget constraint. Extending our algorithms to handle such constraints with our formulation is an interesting future work.

Acknowledgments

The authors would like to thank the anonymous reviewers for their comments. This work has been partially supported by the following grants: DTRA Grant HDTRA1-11-1-0016, DTRA CNIMS Contract HDTRA1-11-D-0016-0010, US National Science Foundation Career CNS 0845700, US National Science Foundation ICESCCF-1216000, US National Science Foundation NETSE Grant CNS-1011769, US National Science Foundation DIBBS Grant ACI-1443054, US National Science Foundation Grant IIS-1353346, NEH Grant HG-229283-15, Maryland Procurement Office under contract H98230-14-C-0127, and a Facebook faculty gift. This work is also supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior National Business Center (DoI/NBC) contract number D12PC000337, the US Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Any opinions, findings, and conclusions or recommendations express in this material are those of the author(s) and do not necessarily reflect the views of the respective funding agencies.

References

[1] J. Medlock and A. P. Galvani, "Optimizing influenza vaccine distribution," *Science*, vol. 325, pp. 1705–1708, 2009.
[2] S. Eubank, et al., "Modelling disease outbreaks in realistic urban social networks," *Nature*, vol. 429, no. 6988, pp. 180–184, May 2004.
[3] D. Z. Roth and B. Henr, "Social distancing as a pandemic influenza prevention measure: Evidence review," National Collaborating Centre for Infectious Diseases, 2011.
[4] E. Shim, "Optimal strategies of social distancing and vaccination against seasonal influenza," *Math. Biosciences Eng.*, vol. 10, no. 5/6, pp. 1615–1634, 2013.
[5] R. Cohen, S. Havlin, and D. ben Avraham, "Efficient immunization strategies for computer networks and populations," *Phys. Rev. Lett.*, vol. 91, no. 24, Dec. 2003, Art. no. 247901.
[6] B. A. Prakash, L. A. Adamic, T. J. Iwashyna, H. Tong, and C. Faloutsos, "Fractional immunization in networks," in *Proc. SIAM Int. Conf. Data Mining*, 2013, pp. 659–667.
[7] H. Tong, B. A. Prakash, C. E. Tsourakakis, T. Eliassi-Rad, C. Faloutsos, and D. H. Chau, "On the vulnerability of large graphs," in *Proc. IEEE Int. Conf. Data Mining*, 2010, pp. 1091–1096.
[8] R. M. Anderson and R. M. May, *Infectious Diseases of Humans*. London, U.K.: Oxford Univ. Press, 1991.
[9] N. Bailey, *The Mathematical Theory of Infectious Diseases and Its Applications*. London, U.K.: Griffin, 1975.
[10] D. Kempe, J. Kleinberg, and E. Tardos, "Maximizing the spread of influence through a social network," in *Proc. 9th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2003, pp. 137–146.
[11] E. B. Khalil, B. Dilkina, and L. Song, "Scalable diffusion-aware optimization of network topology," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2014, pp. 1226–1235.
[12] B. A. Prakash, D. Chakrabarti, M. Faloutsos, N. Valler, and C. Faloutsos, "Threshold conditions for arbitrary cascade models on arbitrary networks," *Knowl. Inf. Syst.*, vol. 33, pp. 549–575, 2012.
[13] S. Saha, A. Adiga, B. A. Prakash, and A. K. S. Vullikanti, "Approximation algorithms for reducing the spectral radius to control epidemic spread," in *Proc. SIAM Int. Conf. Data Mining*, 2015, pp. 568–576.
[14] D. Gruhl, R. V. Guha, D. Liben-Nowell, and A. Tomkins, "Information diffusion through blogspace," in *Proc. 13th Int. Conf. World Wide Web*, May 2004, pp. 491–501.
[15] R. Kumar, J. Novak, P. Raghavan, and A. Tomkins, "On the bursty evolution of blogspace," in *Proc. 12th Int. Conf. World Wide Web*, 2003, pp. 568–576.
[16] S. Bikhchandani, D. Hirshleifer, and I. Welch, "A theory of fads, fashion, custom, and cultural change in informational cascades," *J. Political Economy*, vol. 100, no. 5, pp. 992–1026, 1992.
[17] J. Goldenberg, B. Libai, and E. Muller, "Talk of the network: A complex systems look at the underlying process of word-of-mouth," *Marketing Lett.*, vol. 12, pp. 211–223, 2001.
[18] E. M. Rogers, *Diffusion of Innovations*, 5th ed. New York, NY, USA: Free Press, Aug. 2003.

[19] J. Leskovec, L. A. Adamic, and B. A. Huberman, "The dynamics of viral marketing," in *Proc. 7th ACM Conf. Electron. Commerce*, 2006, pp. 228–237.
[20] E. E. Papalexakis, T. Dumitras, D. H. P. Chau, B. A. Prakash, and C. Faloutsos, "Spatio-temporal mining of software adoption & penetration," in *Proc. IEEE/ACM Int. Conf. Advances Social Netw. Anal. Mining*, 2013, pp. 878–885.
[21] H. W. Hethcote, "The mathematics of infectious diseases," *SIAM Rev.*, vol. 42, pp. 599–653, 2000.
[22] A. G. McKendrick, "Applications of mathematics to medical problems," *Proc. Edinburgh Math. Soc.*, vol. 44, pp. 98–130, 1925.
[23] M. Granovetter, "Threshold models of collective behavior," *Amer. J. Sociology*, vol. 83, pp. 1420–1443, 1978.
[24] D. J. Watts, "A simple model of global cascades on random networks," *Proc. Nat. Academy Sci. USA*, vol. 99, no. 9, pp. 5766–5771, 2002.
[25] D. Centola and M. Macy, "Complex contagions and the weakness of long ties," *Amer. J. Sociology*, vol. 113, no. 3, pp. 702–734, 2007.
[26] A. Ganesh, L. Massoulie, and D. Towsley, "The effect of network topology on the spread of epidemics," in *Proc. IEEE INFOCOM*, 2005, pp. 1455–1466.
[27] J. Aspnes, K. Chang, and A. Yampolskiy, "Inoculation strategies for victims of viruses and the sum-of-squares partition problem," in *Proc. 16th Annu. ACM-SIAM Symp. Discr. Algorithms*, 2005, pp. 43–52.
[28] C. J. Kuhlman, G. Tuli, S. Swarup, M. V. Marathe, and S. Ravi, "Blocking simple and complex contagion by edge removal," in *Proc. IEEE 13th Int. Conf. Data Mining*, 2013, pp. 399–408.
[29] H. Tong, B. A. Prakash, T. Eliassi-Rad, M. Faloutsos, and C. Faloutsos, "Gelling, and melting, large graphs by edge manipulation," in *Proc. 21st ACM Int. Conf. Inf. Knowl. Manage.*, 2012, pp. 245–254.
[30] P. V. Mieghem, et al., "Decreasing the spectral radius of a graph by link removals," *Phys. Rev. E*, vol. 84, 2011, Art. no. 016101.
[31] C. Budak, D. Agrawal, and A. E. Abbadi, "Limiting the spread of misinformation in social networks," in *Proc. 20th Int. Conf. World Wide Web*, 2011, pp. 665–674.
[32] X. He, G. Song, W. Chen, and Q. Jiang, "Influence blocking maximization in social networks under the competitive linear threshold model," in *Proc. SIAM Int. Conf. Data Mining*, 2012, pp. 463–474.
[33] Y. Zhang and B. Prakash, "DAVA: Distributing vaccines over large networks under prior information," in *Proc. SIAM Int. Conf. Data Mining*, 2014, pp. 46–54.
[34] M. Richardson and P. Domingos, "Mining knowledge-sharing sites for viral marketing," in *Proc. 8th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2002, pp. 61–70.
[35] M. Eftekhar, Y. Ganjali, and N. Koudas, "Information cascade at group scale," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2013, pp. 401–409.
[36] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. S. Glance, "Cost-effective outbreak detection in networks," in *Proc. 13th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2007, pp. 420–429.
[37] T. Lappas, E. Terzi, D. Gunopulos, and H. Mannila, "Finding effectors in social networks," in *Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2010, pp. 1059–1068.
[38] M. Purohit, B. A. Prakash, C. Kang, Y. Zhang, and V. Subrahmanian, "Fast influence-based coarsening for large networks," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2014, pp. 1296–1305.
[39] T. Soma, N. Kakimura, K. Inaba, and K.-i. Kawarabayashi, "Optimal budget allocation: Theoretical guarantee and efficient algorithm," in *Proc. 31st Int. Conf. Mach. Learn.*, 2014, pp. 351–359.
[40] M. Minoux, "Accelerated greedy algorithms for maximizing submodular set functions," in *Optimization Techniques*. Berlin, Germany: Springer, 1978, pp. 234–243.
[41] L. Lu and X. Peng, "Spectra of edge-independent random graphs," *Electron. J. Combinatorics*, vol. 20, no. 4, 2013, Art. no. P27.
[42] S. Sahni, "Computationally related problems," *SIAM J. Comput.*, vol. 3, no. 4, pp. 262–279, 1974.
[43] A. Goyal, F. Bonchi, and L. V. Lakshmanan, "Learning influence probabilities in social networks," in *Proc. 3rd ACM Int. Conf. Web Search Data Mining*, 2010, pp. 241–250.
[44] A. F. McDaid, T. B. Murphy, N. Friel, and N. J. Hurley, "Improved Bayesian inference for the stochastic block model with application to large networks," *Comput. Statist. & Data Anal.*, Elsevier, vol. 60, pp. 12–31, 2013.
[45] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *J. Statistical Mech.: Theory Experiment*, vol. 2008, no. 10, 2008, Art. no. P10008.

Yao Zhang received the bachelor's and master's degrees in computer science from Nanjing University, China. He is working toward the PhD degree in the Department of Computer Science, Virginia Tech. His current research interests include graph mining and social network analysis with focus on understanding and managing information diffusion in networks. He has published several papers in top conferences and journals such as the *Knowledge Discovery in Databases*, ICDM, SDM, and the *Transactions on Knowledge Discovery from Data*.

Abhijin Adiga received the PhD degree from the Department of Computer Science & Automation, Indian Institute of Science, in 2011. He is a research assistant professor in the Network Dynamics and Simulation Science Laboratory, Biocomplexity Institute of Virginia Tech. His interests include network science, modeling, algorithms, combinatorics, and game theory with current focus on dynamical processes over networks and design & implementation of complex simulation systems.

Sudip Saha received the BSc degree in computer science and engineering from the Bangladesh University of Engineering and Technology, in 2006, and the MS degree in computer science from the University of Memphis, Tennessee, in 2010. He is currently working toward the PhD degree in computer science at Virginia Tech. His research interests include graph mining, game theory, and cybersecurity.

Anil Vullikanti received the undergraduate degree from the Indian Institute of Technology, Kanpur, and the PhD degree from the Indian Institute of Science, Bangalore. He is an associate professor in the Department of Computer Science and the Biocomplexity Institute of Virginia Tech. He was a post-doctoral researcher with the Max-Planck Institute for Informatics, and a technical staff member with the Los Alamos National Laboratory. His research interests include the broad areas of approximation and randomized algorithms and dynamical systems, and their applications to computational epidemiology and the modeling, simulation and analysis of socio-technical systems.

B. Aditya Prakash received the BTech (computer science) degree from the Indian Institute of Technology (IIT)—Bombay, in 2007, and the PhD degree from the Computer Science Department, Carnegie Mellon University, in 2012. He is an assistant professor in the Computer Science Department, Virginia Tech. His research interests include data mining, applied machine learning, and databases, with emphasis on big-data problems in large real-world networks and time-series. He has published more than 40 refereed papers in major venues, holds two US patents and has given two tutorials (VLDB 2012 and ECML/PKDD 2012) at leading conferences. His work has also received a best paper award, two best-of-conference selections (CIKM 2012, ICDM 2012, and ICDM 2011), and multiple travel awards. His work has been funded through grants/gifts from US NSF, DoE, NSA, NEH and from companies like Symantec. He received a Facebook Faculty Gift Award in 2015. He is also an affiliated faculty member in the Discovery Analytics Center, Virginia Tech.