Controlling Propagation at Group Scale on Networks

Yao Zhang*, Abhijin Adiga†, Anil Vullikanti*,† and B. Aditya Prakash*
*Department of Computer Science, Virginia Tech
†NDSSL, Virginia Bioinformatics Institute, Virginia Tech
Email: {yaozhang, badityap}@cs.vt.edu, {abhijin, akumar}@vbi.vt.edu

***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 best choose communities in social networks 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 novel so-called 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.

I. 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. However, 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 allocation within each group, which motivates our paper. We study both vaccination problems (which can be modeled by node removal from the network) and social distancing (which can be modeled by edge removal).

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 [8]. Analogous to the public-health case, we can control such processes by 'immunization' via blocking users or preventing some interactions. Past work has studied individual-level based immunization algorithms for the LT model [9]. However, it is more realistic to issue a warning bulletin on group pages, and members within groups can get 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:

(a) 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.

(b) Effective Algorithms: We develop efficient theoretical and practical algorithms for the four problem classes we consider. We find that diverse kinds of techniques are needed for these problems—submodular function maximization on an integer lattice, quadratic programming and semidefinite programming. Our algorithms leverage prior techniques for analyzing contagion processes, e.g., [10], [9], [6], but require non-trivial extensions.

(c) 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.

II. Our Problem Formulations

Table I lists the main symbols we use throughout the paper. Here we assume our graph $G(V, E)$ is directed and weighted.

We give two different sets of problems which cover a wide range of contagion-like processes both threshold-based and

TABLE I. 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 $(x_1, \cdots, x_n)$ 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., $\arg\min_{\mathbf{x}} \lambda(\mathbf{M}_{\mathbb{E}}(\mathbf{x}))$

cascade-style. In addition, all our problems have been carefully formulated to be seamless generalizations of the corresponding individual-level problems.

Our first set of problems are based on the LT model which is a well-known model for social media and complex propagations [8] 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 [9].

Our second set of problems are based on the spectral radius formulation [7], [11] following the fundamental SIR and SIS models that contain the popular IC model [8] as a special case (more details in Section II-B). Recent results [10], [12] have shown that the spectral radius is connected to the reproduction number in epidemiology and determines the phase-transition ('epidemic threshold') between epidemic/non-epidemic regimes in a very large range of cascade-style models (including SIR/SIS/IC models).

We refer to both node and edge level interventions as immunization. For a graph $G(V, E)$, we assume that the edge (node) set is partitioned into groups $C = \{C_1, \ldots, C_n\}$ for the edge (node) immunization problems. Note that we assume there are no overlaps among 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 is to find an allocation that controls the diffusion process most effectively.

A. Problem Definition under LT model

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 $\sum_{w\in N_u^1} p_{wu} \geq \theta_u$ where N_u^1 is the set of active neighbors of v 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 $\sigma_{C,A}(\mathbf{x})$ ($\mathbb{Z}^n \to \mathbb{R}$) 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. $|\mathbf{x}| \leq m$.*

Note that 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 [9].

B. Problem Definition for spectral radius

As mentioned before the largest eigenvalue of the adjacency matrix of a network (a.k.a., spectral radius), λ, is an important metric which can be linked to the reproduction number and the epidemic threshold τ of a graph G (i.e. $\tau \propto \frac{1}{\lambda}$) for a broad-range of cascade-style epidemic models [10] including SIR ('mumps-like' which generalizes the IC model), SIS ('flu-like'), SEIS (with incubation period) and so on. An epidemic will be quickly extinguished given a small enough λ. Tong et al [7], [11] 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], [11]; 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. $|\mathbf{x}| \leq m$.*

Problems 3 and 4 are NP-hard too—their special cases, individual-level immunizations are NP-hard [7], [11].

III. PROPOSED METHODS

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

A. Edge Deletion under LT model

The function $f(\mathbf{x})$ we are trying to optimize in Problem 1 is defined over an integer lattice, and is not a simple set function. Our approach is to 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., [13], who show 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 diminishing return property. Besides, there are multiple non-equivalent definitions of the diminishing return property, as observed in [13]. We show below in Theorem 1 that a greedy algorithm gives an $(1-1/e)$-factor approximation to a function $f(\mathbf{x})$ satisfying the properties (P_1), (P_2) and (P_3) above. It is not clear whether the analysis of [13] implies a similar bound for the kind of functions $f(\mathbf{x})$ we need to consider here.

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 $\sum_i x_i \leq m$.

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** $\mathbf{x}$

Proof: The proof is in the online appendix[1]. ■

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 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, $\sum_{w \in N_v} p_{wv} = \sum_{w \in N(v) \cap A} p_{wv} = p_{sv}$. 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 1: 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 [8], 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

[1] Appendix: http://people.cs.vt.edu/~yaozhang/group-immu/.

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

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, $\rho = \prod_{i \neq k} \frac{1}{\binom{c_i}{x_i}}$.

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

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

$$= \frac{\rho x_k!(c_k - x_k - 1)!}{c_k!} \sum_{R \in \mathcal{R}(\mathbf{x})} \sum_{e \in C_k \setminus R} \hat{\gamma}(R) - \hat{\gamma}(R \cup \{e\}) . \quad (1)$$

Similarly, we have $\sigma_C(\hat{G}, \mathbf{x}') - \sigma_C(\hat{G}, \mathbf{x}' + \mathbf{e}_k)$

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

From [9, 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 $\rho' = \prod_{i \neq j,k} \frac{1}{\binom{c_i}{x_i}}$. From eqn. (1), we have $\sigma(\hat{G}, \mathbf{x}) - \sigma(\hat{G}, \mathbf{x} + \mathbf{e}_k)$. $\sigma_C(\mathbf{x}') - \sigma_C(\mathbf{x}' + \mathbf{e}_k)$

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

Again from [9], $(c_j - x_j - 1) \sum_e \hat{\gamma}(R) - \hat{\gamma}(R \cup \{e\}) \geq \sum_{e_j} \sum_e \hat{\gamma}(R \cup \{e_j\}) - \hat{\gamma}(R \cup \{e_j, e\})$. Hence proved. ■

Algorithm 1 provides a simple greedy algorithm. In Algorithm 1, to estimate $\sigma_C(\mathbf{x})$, we can 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. [8] show that $\gamma_C(R)$ can be estimated by sampling from the set of live-edge graphs. 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.

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_X^s where s is the root, and $r(u, T_X^s) = |\{v | v \in subtree(u)\}|$, i.e., the number of nodes that are under the subtree of u in T_X^s. 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 $r(u, T_X^s)$ in parallel for all nodes in all the live graphs (Line 1-3). After that we greedily select m vaccines (Line 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 $\Delta_{C_i,s}(\mathbf{x} + \mathbf{e_i}) = \sum_{e(u,v) \in T_X^s} r(s, T_X^s) - r(s, T_X^s \setminus e)$, i.e., we randomly pick one edge from each group for each live-edge graph, then sum their marginal losses up over T_X^s as C_i's marginal loss. Note that $r(s, T_X^s) - r(s, T_X^s \setminus e) = r(v, T_X^s) + 1$, where node v is the endpoint of e [9]. We pick the group C^* with the maximum marginal loss. Finally we removed the edge that has been picked, and update $r(u, T_X^s)$ in parallel (Line 11-13). There are two cases to update T_X^s if $e(u, v) \in T_X^s$: (1) for v's children, we can remove them because it is not reachable from s; (2) for any ancestor a of v, $r(a, T_X^s \setminus e) = r(a, T_X^s) - r(v, T_X^s) - 1$, 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 $\sigma_C(\mathbf{x})$.

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 $\mathcal{M} = \{T_{X_1}^I, \ldots, T_{X_{|\mathcal{M}|}}^I\}$
3: For each T_X^I, calculate $r(u, T_X^I)$ for all nodes (in parallel)
4: Set $\mathbf{x} = \mathbf{0}$
5: **for** $j = 1$ to m **do**
6: **for** each T_X^I and C_i **do**
7: pick an edge $e_X^{C_i}$ at random for C_i and T_X^I
8: **end for**
9: $C^* = \arg\max_{C_i} \sum_{e_X^{C_i} \in T_X^I} (r(I, T_X^I) - r(I, T_X^I \setminus e_X^{C_i}))$
10: $x_{C^*} = x_{C^*} + 1$
11: **for** each T_X^I **do**
12: If $e_X^{C^*}(u, v) \in T_X^I$, remove edge $e_X^{C_i}$ and update $r(n, T_X^I)$ for node n (in parallel)
13: **end for**
14: **end for**
15: **return** $\mathbf{x}$

Running Time of GREEDY-LT. Calculating all $r(u, T_X^I)$ costs $O(|\mathcal{M}||V|)$ time since we can traverse T_X^I once to get all values of $r(u, T_X^I)$. 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 $r_i(u, T_X^I)$ in parallel. In addition, since T_X^I is tree, the increasing difference property still holds, hence we can accelerate GREEDY-LT by "lazy evaluation" [14], [15] as well.

B. 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 III-A. 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 III-A, as discussed below.

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).

Proof: (Sketch) Our proof follows on the same lines as the proof of Lemma 1. As a first step, we need to prove that $\gamma'_C(R)$ is monotone non-increasing and supermodular, where $\gamma'_C(R)$ is the expected number of infected nodes at the end of the LT process after R is removed. We show that this follows from the corresponding property for the function $\gamma_C(\cdot)$, as used in Lemma 1.

First, for any node set $S \subset V$, we define edge set $E_S = \{e(i,j) \mid i \in S \vee j \in S\}$. Let $S, T \subset V$ with $S \subseteq T$. Then, $E_S \subseteq E_T$, so that $\gamma'_C(S) = \gamma_C(E_S) \leq \gamma_C(E_T) = \gamma'_C(T)$. This proves $\gamma'_C(\cdot)$ is monotonically decreasing. Next, we prove below that $E_{S\cap T} \subseteq E_S \cap E_T$ and $E_{S\cup T} \subseteq E_S \cup E_T$. For the former, observe that for any edge $(i,j) \in E_{S\cap T}$, we must have either $i \in S \cap T$ or $j \in S \cap T$. From the definition of E_S and E_T, it follows that $(i,j) \in E_S \cap E_T$, so that $E_{S\cap T} \subseteq E_S \cap E_T$. Similarly, if $(i,j) \in E_{S\cup T}$, we have either $i \in S \cup T$ or $j \in S \cup T$. This implies that $(i,j) \in E_S \cup E_T$.

Hence, $\gamma_C(E_S \cap E_T) \leq \gamma_C(E_{S\cap T})$ and $\gamma_C(E_S \cup E_T) \leq \gamma_C(E_{S\cup T})$. According to the supermodularity of $\gamma_C(\cdot)$, $\gamma_C(E_S)+\gamma_C(E_T) \leq \gamma_C(E_S \cup E_T)+\gamma_C(E_S \cap E_T)$, therefore, $\gamma'_C(S) + \gamma'_C(T) = \gamma_C(E_S) + \gamma_C(E_T) \leq \gamma_C(E_S \cup E_T) + \gamma_C(E_S \cap E_T) \leq \gamma_C(E_{S\cup T}) + \gamma_C(E_{S\cap T}) = \gamma'_C(S \cup T) + \gamma'_C(S \cap T)$, which proves $\gamma'_C(\cdot)$ is supermodular.

Following the equivalence between the LT model and the directed percolation process [8], 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, 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 of nodes is removed. The proof then follows as in Lemma 1, using the supermodularity of $\gamma'_C(\cdot)$. ∎

Lemma 2 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., $r(I, T_X^I) - r(I, T_X^I \setminus v) = r(v, T_X^I) + 1$. Hence, the updating process is the same as the edge version of GREEDY-LT.

C. Edge Deletion for Spectral Radius

We propose two algorithms for Problem 3, edge immunization based on spectral radius. While the first one is an SDP formulation based on the actual eigendrop, the second algorithm is an LP-based method which uses an approximation of the eigendrop. The time complexity of SDP ($O(|V|^4\text{polylog}(|V|))$) was one of the primary motivations to come up with the much faster LP heuristic (its complexity depends only on the number of groups, not graph size). On the other hand, SDP has proven performance guarantee, and is not merely of theoretical interest; it can be used as a baseline to assess the performance of faster heuristics on smaller networks.

1) An SDP approach: Let $G(V, E)$ be a graph whose edge set is partitioned into n groups $C_1, \cdots, C_n$. 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 3.1: 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.

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.

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

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

Relating $\lambda_{\mathbb{E}}^{\min}$ to $\lambda(\mathbf{M}_{\mathbb{E}}(\mathbf{x}_{\min}))$. One can use the following result by Lu and Peng [16] to bound $\lambda_{\mathbb{E}}(\mathbf{x})$ with respect to $\lambda(\mathbf{M}_{\mathbb{E}}(\mathbf{x}))$.

Theorem 2 ([16]): 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

$$|\lambda_i(\mathbf{M}(H)) - \lambda_i(\mathbf{M}_{\mathbb{E}}(H))| \leq (2 + o(1))\sqrt{\Delta_{\mathbb{E}}(H)},$$

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 $G(\mathbf{x}_{\min})$.

Lemma 3: If $\mathbf{x}_{\min}$ is such that $\Delta_{\mathbb{E}}(\mathbf{x}_{\min}) \gg \log^4 |V|$, then, $\lambda_{\mathbb{E}}^{\min} \leq \lambda(\mathbf{M}_{\mathbb{E}}(\mathbf{x}_{\min})) + (2 + o(1))\sqrt{\Delta_{\mathbb{E}}(\mathbf{x}_{\min})}$+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 $G(\mathbf{x}_{\min})$, $\lambda(\mathbf{M}(\mathbf{x}_{\min})) \leq z$ almost surely. In fact, for $\Delta_{\mathbb{E}}(\mathbf{x}_{\min}) \gg \log^4 |V|$, it can be shown that $\Pr\big(\lambda(\mathbf{M}(\mathbf{x}_{\min})) \geq z\big) \leq 1/|V|$ (see [16, proof of Theorem 6]). Noting that $\lambda(\mathbf{M}(\mathbf{x}_{\min})) \leq \lambda(\mathbf{M})$,

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

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

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

2) A Method Based on Approximate 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 $\mathbf{u} = (u_1, \ldots, u_i, \ldots)$ [11]. 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.

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

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. Hence, if the number of groups is small, this algorithm is very fast.

D. 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 III-C. 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

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

Applying the above to (8),

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

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,

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

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

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

where, $\mathbf{Q}_{aa} = 2\beta_a$ and for $a \neq b$, $\mathbf{Q}_{ab} = 2\Gamma_{ab}$ and $\mathbf{c}_a = -\alpha_a$. If $\mathbf{Q}$ is not semi-definite, the problem is NP-Hard [17]. 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 $\hat{\mathbf{Q}}$ can be solved in polynomial time using the ellipsoid method [17].

Lemma 4: $|\widehat{\phi}(\mathbf{x}_{\widehat{\mathbf{Q}}}) - \phi(\mathbf{x}_{\mathbf{Q}})| \leq \frac{n}{2} \cdot \|\mathbf{Q} - \widehat{\mathbf{Q}}\|_F$, where n is the number of groups in the graph.

Proof: Let $\mathbf{x}_{\mathbf{Q}}$ and $\mathbf{x}_{\widehat{\mathbf{Q}}}$ correspond to the best allocation vectors corresponding to $\mathbf{Q}$ and $\widehat{\mathbf{Q}}$ respectively. Let $\widehat{\phi}(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T \widehat{\mathbf{Q}} \mathbf{x} + \mathbf{c}^T \mathbf{x}$. Then,

$$\begin{aligned}\widehat{\phi}(\mathbf{x}_{\widehat{\mathbf{Q}}}) - \phi(\mathbf{x}_{\mathbf{Q}}) &= \Big(-\frac{1}{2}\mathbf{x}_{\widehat{\mathbf{Q}}}^T \widehat{\mathbf{Q}} \mathbf{x}_{\widehat{\mathbf{Q}}} - \mathbf{c}^T \mathbf{x}_{\widehat{\mathbf{Q}}}\Big) \\ &\qquad - \Big(-\frac{1}{2}\mathbf{x}_{\mathbf{Q}}^T \mathbf{Q} \mathbf{x}_{\mathbf{Q}} - \mathbf{c}^T \mathbf{x}_{\mathbf{Q}}\Big) \\ &\leq \Big(-\frac{1}{2}\mathbf{x}_{\widehat{\mathbf{Q}}}^T \widehat{\mathbf{Q}} \mathbf{x}_{\widehat{\mathbf{Q}}} - \mathbf{c}^T \mathbf{x}_{\widehat{\mathbf{Q}}}\Big) \\ &\qquad - \Big(-\frac{1}{2}\mathbf{x}_{\widehat{\mathbf{Q}}}^T \mathbf{Q} \mathbf{x}_{\widehat{\mathbf{Q}}} - \mathbf{c}^T \mathbf{x}_{\widehat{\mathbf{Q}}}\Big) \\ &= \Big(\frac{1}{2}(\mathbf{x}_{\widehat{\mathbf{Q}}}^T (\mathbf{Q} - \widehat{\mathbf{Q}}) \mathbf{x}_{\widehat{\mathbf{Q}}}\Big) \\ &\leq \frac{n}{2} \mathbf{y}^T (\mathbf{Q} - \widehat{\mathbf{Q}}) \mathbf{y}\end{aligned}$$

where $\mathbf{y}$ is some unit vector. The last expression follows from the fact that $\mathbf{x}_{\widehat{\mathbf{Q}}}^T \mathbf{x}_{\widehat{\mathbf{Q}}} \leq n$. Finally, we note that $|\mathbf{y}^T (\mathbf{Q} - \widehat{\mathbf{Q}}) \mathbf{y}| \leq \|\mathbf{Q} - \widehat{\mathbf{Q}}\|_F$. Hence, proved. ∎

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.

IV. Empirical Study

A. Experimental Setup

We implemented the algorithms in Python[2] , and conducted the experiments using a 4 Xeon E7-4850 CPU with 512GB of 1066Mhz main memory.

Datasets. We run our experiments on multiple datasets.

1). Stochastic Block Model (SBM) [18] is a well-known graph model to generate synthetic graph with groups. We generate a graph with 20 groups **1500** nodes **5000** edges.

2). Protein[3] is a protein-protein interaction network in budding yeast with **2361** nodes and **7182** edges. There are 13 classes of proteins, which are naturally treated as groups.

3). OregonAS[4] is the Oregon AS router graph collected from the Oregon router views, which contains **10670** nodes and **22002** edge, and the group here are based on router conductivities. We use Louvain [19], a fast community detection algorithm to specify the groups.

4). YouTube[5] a friendship networks in which users can form groups. We create an induced graph by selecting nodes that are in the top 5000 communities, which contains about **50K** nodes and **450K** edges.

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]. Portland contains **0.5million** nodes and **1.6million** edges, while Miami has **0.6million** nodes and **2.1million** edges. We divided people by ages ranging from 0-90 (hence 91 groups in both networks).

Settings. For LT model, we uniformly randomly choose 1% nodes as the infected nodes (seeds) at the start. And we use the same method in [9] 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 500 live-edge graphs in our algorithm for LT model. For robustness, each data point we show is the mean of 1000 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 a group C_t, then $e \in C_t$, otherwise it belongs to group $C_{ij} = \{e_{uv}|u \in C_j, v \in C_j\}$.

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], [11], [20].

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 [21] 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], [11], even as a baseline for LT model [9]. 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 $|u_a u_b|$ [11]. 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 $u_{C_i}/\sum_{C_k\in C} u_{C_k}$.

Note that we do not compare and run the individual based immunization methods [9], [7] "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. 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).

B. 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.

Note that we have given the time complexity of each algorithm. Some of our algorithms, e.g., the one based on 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.

1) Performance: Figure 1(a), (b) and (c) show experimental results under LT model for group edge deletion, while (d), (e) and (f) demonstrate 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 Figure 1(a), (b) and (c), Greedy-LT performs pretty well for edge deletion compared with other competitors, e.g., in YouTube, Greedy-LT can reduce about 25% 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, Greedy-LT performs even better: it reduces more than 30% of the infection given the maximum budgets.

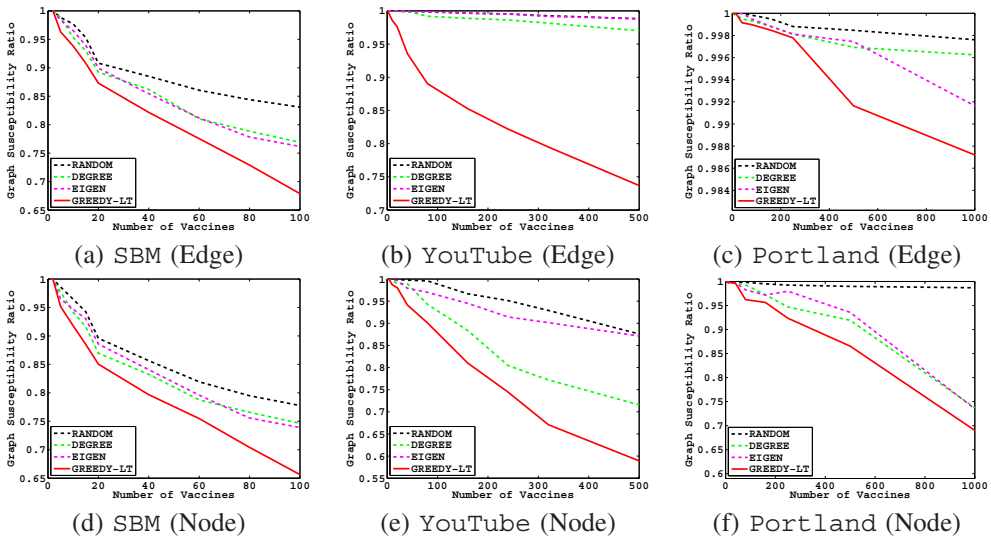
Figure 2(a), (b) and (c) show experimental results of edge verion of group immunization for spectral radius, while

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

[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 (Edge) (b) YouTube (Edge) (c) Portland (Edge)

(d) SBM (Node) (e) YouTube (Node) (f) Portland (Node)

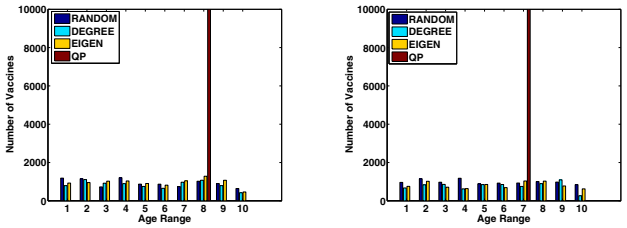
Fig. 1. Effectiveness for LT model various Real Datasets. (a),(b) and (c): edge deletion; (d), (e) and (f): node deletion. Graph susceptibility ratio ($\frac{footprint\ when\ vaccines\ are\ given}{footprint\ without\ giving\ vaccines}$) vs. number of vaccines. Lower is better. GREEDY-LT consistently outperforms other baseline algorithms for both node and edge deletion.

(c), (d) and (e) demonstrate the results for node deletion. In all networks, SDP, LP and QP consistently outperform other competitors. SDP gives the best results for Protein, however, it is not scalable to large networks like YouTube and Portland. LP for edge deletion and QP for node deletion, perform very well for large networks. For edge deletion, RANDOM, DEGREE and EIGEN cannot decrease more than 10% of the first eigenvalue in YouTube when $5k$ vaccines are given to groups, while LP can reduce more than 20% of the eigenvalue. For node deletion, 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 Figure 1: given same vaccines to both edge and node, node removal can get a larger decrease of the spectral radius.

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 [19] to find different number of groups. For YouTube, we randomly merge ground true communities to form smaller size of groups.

Figure 3(a) and (b) 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. Figure 3(c) and (d) 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% of the infection.

3) 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. Figure 4 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, for the Portland and Miami networks, respectively. This contrasts with CDC recommendations, and the strategy proposed by Medlock et al. [1]. This might be because these results do not use the detailed network structure. We believe this is an interesting result which merits further study.



(a) Portland (b) Miami

Fig. 4. Vaccine Distributions for Portland and Miami (Budget=10000). Number of vaccines vs. 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.)

V. RELATED WORK

In general, there has been a lot of interest in studying dynamic processes on large graphs like (a) blogs and propagations [22], (b) information cascades [23]; (c) marketing and product penetration [24] and (d) malware prediction [25].

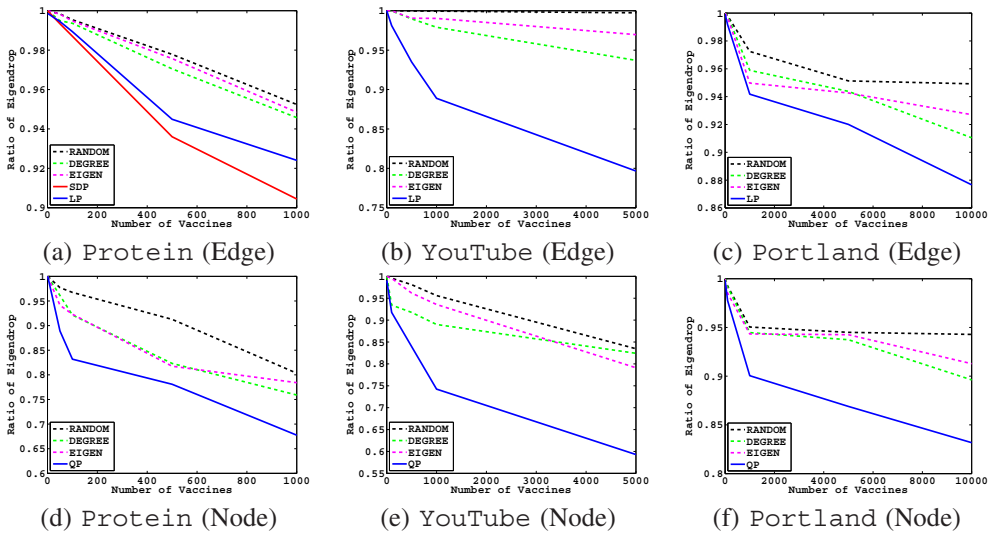


Fig. 2. Effectiveness for the change of the first eigenvalue various Real Datasets. (a),(b) and (c): edge deletion; (d), (e) and (f): node deletion. Eigendrop ratio ($\frac{\lambda'_G}{\lambda_G}$) vs. number of vaccines (λ'_G is the expected eigenvalue after allocating vaccines). Lower is better. SDP, LP and QP consistently outperform other baseline algorithms for both node and edge deletion.

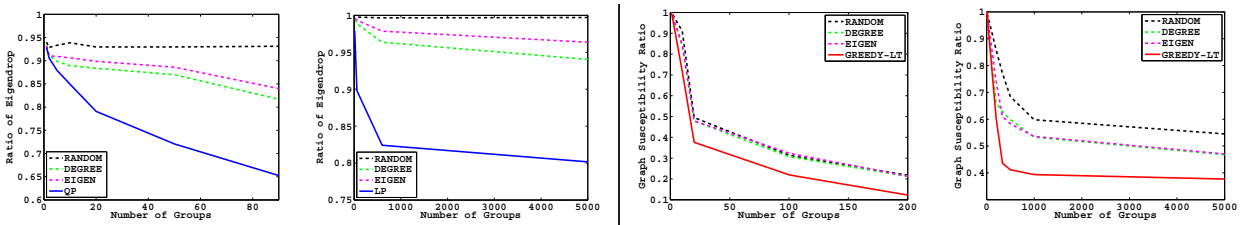


Fig. 3. (a) and (b). Eigendrop ratio vs. number of groups. (c) and (d): Graph susceptibility ratio vs. number of groups. Lower is better. Our algorithms consistently outperform other baseline algorithms as the number of groups changes as well as the size of groups changes.

These dynamic processes are all closely related to virus propagation. In this ection, we review related work mainly from three areas: epidemiology, immunization and other 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 [26], [27]. Most work in epidemiology is focused on *homogeneous models*. Much work has gone into in finding epidemic thresholds for network models (minimum virulence of a virus which results in an epidemic) for a variety of virus propagation models [28], [10].

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. [29] developed inoculation strategies for victims of viruses under random starting points. Tong et al. [7], [11], Van Miegham et al. [21], 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 [30], Khalil et al. [9] and Zhang et al. [31]. 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 Optimization Problems: Other diffusion based optimization problems include the influence maximization problem, formulated by Kempe et. al. [8] as a combinatorial optimization problem. Recently the paper by Eftekhar et al. [32] studied this problem at group scale. Other such problems where we wish to select a subset of '*important*' vertices on graphs, include 'outbreak detection' [15] and 'finding most-likely culprits of epidemics' [33].

VI. 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 per-

formance 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 and semidefinite programming. 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.

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 direction.

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, NSF Career CNS 0845700, NSF ICESCCF-1216000, NSF NETSE Grant CNS-1011769, NSF DIBBS Grant ACI-1443054, NSF Grant IIS-1353346, Maryland Procurement Office under contract H98230-14-C-0127, and a Facebook faculty gift. 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, 2009.

[2] S. Eubank, H. Guclu, V. S. Anil Kumar, M. V. Marathe, A. Srinivasan, Z. Toroczkai, and N. Wang, "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," 2011.

[4] E. Shim, "Optimal strategies of social distancing and vaccination against seasonal influenza," *Mathematical biosciences and engineering*, vol. 10(5), 2013.

[5] R. Cohen, S. Havlin, and D. ben Avraham, "Efficient immunization strategies for computer networks and populations," *Physical Review Letters*, vol. 91, no. 24, Dec. 2003.

[6] B. A. Prakash, L. A. Adamic, T. J. Iwashyna, H. Tong, and C. Faloutsos, "Fractional immunization in networks," in *Proc. of SDM*, 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 *ICDM*, 2010.

[8] D. Kempe, J. Kleinberg, and E. Tardos, "Maximizing the spread of influence through a social network," in *KDD '03*, 2003.

[9] E. B. Khalil, B. Dilkina, and L. Song, "Scalable diffusion-aware optimization of network topology," in *KDD 2014*. ACM, 2014, pp. 1226–1235.

[10] B. A. Prakash, D. Chakrabarti, M. Faloutsos, N. Valler, and C. Faloutsos, "Threshold conditions for arbitrary cascade models on arbitrary networks," *Knowledge and Information Systems*, 2012.

[11] H. Tong, B. A. Prakash, T. Eliassi-Rad, M. Faloutsos, and C. Faloutsos, "Gelling, and melting, large graphs by edge manipulation," in *Proc. of CIKM*, 2012.

[12] A. J. Ganesh, L. Massoulié, and D. F. Towsley, "The effect of network topology on the spread of epidemics," in *INFOCOM*, 2005, pp. 1455–1466.

[13] T. Soma, N. Kakimura, K. Inaba, and K.-i. Kawarabayashi, "Optimal budget allocation: Theoretical guarantee and efficient algorithm," in *Proceedings of The 31st International Conference on Machine Learning*, 2014, pp. 351–359.

[14] M. Minoux, "Accelerated greedy algorithms for maximizing submodular set functions," in *Optimization Techniques*. Springer, 1978, pp. 234–243.

[15] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. S. Glance, "Cost-effective outbreak detection in networks," in *KDD*, 2007, pp. 420–429.

[16] L. Lu and X. Peng, "Spectra of edge-independent random graphs," *the electronic journal of combinatorics*, vol. 20, no. 4, p. P27, 2013.

[17] S. Sahni, "Computationally related problems," *SIAM Journal on Computing*, vol. 3, no. 4, pp. 262–279, 1974.

[18] A. F. McDaid, B. Murphy, N. Friel, and N. Hurley, "Clustering in networks with the collapsed stochastic block model," *Arxiv preprint arXiv:1203.3083*, Helen Martin 2012.

[19] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, p. P10008, 2008.

[20] S. Saha, A. Adiga, B. A. Prakash, and A. K. S. Vullikanti, "Approximation algorithms for reducing the spectral radius to control epidemic spread." SDM, 2015.

[21] P. V. Mieghem, D. Stevanovic, F. F. Kuipers, C. Li, R. van de Bovenkamp, D. Liu, and H. Wang, "Decreasing the spectral radius of a graph by link removals," *IEEE Transactions on Networking*, 2011.

[22] R. Kumar, J. Novak, P. Raghavan, and A. Tomkins, "On the bursty evolution of blogspace," in *WWW '03*. New York, NY, USA: ACM Press, 2003, pp. 568–576.

[23] J. Goldenberg, B. Libai, and E. Muller, "Talk of the network: A complex systems look at the underlying process of word-of-mouth," *Marketing Letters*, 2001.

[24] E. M. Rogers, *Diffusion of Innovations, 5th Edition*. Free Press, August 2003.

[25] E. E. Papalexakis, T. Dumitras, D. H. P. Chau, B. A. Prakash, and C. Faloutsos, "Spatio-temporal mining of software adoption & penetration," in *IEEE/ACM ASONAM*, Niagara Falls, CA, Aug 2103.

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

[27] H. W. Hethcote, "The mathematics of infectious diseases," *SIAM Review*, vol. 42, 2000.

[28] A. Ganesh, L. Massoulie, and D. Towsley, "The effect of network topology on the spread of epidemics," in *Proc. of INFOCOM*, 2005.

[29] J. Aspnes, K. Chang, and A. Yampolskiy, "Inoculation strategies for victims of viruses and the sum-of-squares partition problem," in *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, ser. SODA '05, 2005, pp. 43–52.

[30] C. Budak, D. Agrawal, and A. E. Abbadi, "Limiting the spread of misinformation in social networks," in *Proc. of WWW*, 2011.

[31] Y. Zhang and B. Prakash, "Dava: Distributing vaccines over large networks under prior information," in *2014 SIAM International Conference on Data Mining (SDM14)*, ser. SDM'14, 2014.

[32] M. Eftekhar, Y. Ganjali, and N. Koudas, "Information cascade at group scale," in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '13. New York, NY, USA: ACM, 2013, pp. 401–409.

[33] T. Lappas, E. Terzi, D. Gunopulos, and H. Mannila, "Finding effectors in social networks," in *Proceedings of the 16th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), Washington, DC*, 2010, pp. 1059–1068.