

Approximation Algorithms for Reducing the Spectral Radius to Control Epidemic Spread

Sudip Saha[§] Abhijin Adiga[§] B. Aditya Prakash[§] Anil Kumar S. Vullikanti[§]

Abstract

The largest eigenvalue of the adjacency matrix of a network (referred to as the spectral radius) is an important metric in its own right. Further, for several models of epidemic spread on networks (e.g., the \mathcal{SIS} model), it has been shown that an epidemic dies out quickly if the spectral radius of the graph is below a certain threshold that depends on the model parameters. This motivates a strategy to control epidemic spread by reducing the spectral radius of the underlying network.

In this paper, we develop a suite of provable approximation algorithms for reducing the spectral radius by removing the minimum cost set of edges (modeling quarantining) or nodes (modeling vaccinations), with different time and quality tradeoffs. Our main algorithm, GreedyWalk, is based on the idea of hitting closed walks of a given length, and gives an $O(\log^2 n)$ -approximation, where n denotes the number of nodes; it also performs much better in practice compared to all prior heuristics proposed for this problem. We further present a novel sparsification method to improve its running time.

In addition, we give a new primal-dual based algorithm with an even better approximation guarantee ($O(\log n)$), albeit with slower running time. We also give lower bounds on the worst-case performance of some of the popular heuristics. Finally we demonstrate the applicability of our algorithms and the properties of our solutions via extensive experiments on multiple synthetic and real networks.

1 Introduction

Given a contact network, which contacts should we remove to contain the spread of a virus? Equivalently, in a computer network, which connections should we cut to prevent the spread of malware? Designing effective and low cost interventions are fundamental challenges in public health and network security. Epidemics are commonly modeled by stochastic diffusion processes, such as the so-called \mathcal{SIS} -like and \mathcal{SIR} -like models on networks (more in Section 2). An important result that highlights the impact of the network structure on the dynamics is that epidemics

die out quickly if $\rho(G) < T$, where $\rho(G)$ is the spectral radius (or the largest eigenvalue) of graph G , and T is a threshold that depends on the disease model [13, 39, 30]. This motivates the following strategy for controlling an epidemic: remove edges (quarantining) or nodes (vaccinating) to reduce the spectral radius below a threshold T . We refer to this as the spectral radius minimization (SRM) problem, with variants depending on whether edges are removed (the SRME problem) or whether nodes are removed (the SRMN problem). Van Mieghem et al. [27] and Tong et al. [37] prove that this problem is NP-complete. They also study two heuristics for it, one based on the components of the ρ -eigenvector (EigenScore) and another based on degrees (ProductDegree). However, no rigorous approximations were known for the SRME or the SRMN problems.

Our main contributions.

1. Lower bounds on the worst-case performance of heuristics: We show that the ProductDegree, EigenScore and Pagerank heuristics (defined formally in Section 2) can perform quite poorly in general. We demonstrate graph instances where these heuristics give solutions of cost $(\frac{n}{T})$ times the optimal, where n is the number of nodes in the graph.
2. Provable approximation algorithms: We present two bicriteria approximation algorithms for the SRME and SRMN problems, with varying approximation quality and running time tradeoffs. Our ρ algorithm, GreedyWalk, is based on hitting closed walks in G . We show this algorithm has an approximation bound of $O(\log n \log \rho)$ times optimal for the cost of edges removed, while ensuring that the spectral radius becomes at most $(1 + \epsilon)$ times the threshold, for arbitrarily small ϵ (here ρ denotes the maximum node degree in the graph). We also design a variant, GreedyWalkSparse, that performs careful sparsification of the graph, leading to similar asymptotic guarantees, but better running time, especially when the threshold T is small. We then develop algorithm PrimalDual, which improves this approximation bound to an $O(\log n)$ using a more sophisticated primal-dual approach, at the

[§]NDSSL, Virginia Bioinformatics Institute, Virginia Tech.

[§]Department of Computer Science, Virginia Tech.

Email: {ssaha, abhijin, akumar}@vbi.vt.edu, badityap@cs.vt.edu

expense of a slightly higher (but polynomial) running time.

3. Empirical analysis: We conduct an extensive experimental evaluation of GreedyWalk, a simplified version of PrimalDual and different heuristics that have been proposed for epidemic containment on a diverse collection of synthetic and real networks. For SRME, these heuristics involve picking edges $e = (i, j)$ in non-increasing order of some kind of score; the specific heuristics we compare include: (i) Product-Degree, (ii) EigenScore, (iii) LinePagerank, and (iv) Hybrid, which pick the edge based on either the eigenscore or the product-degree ordering, depending on the maximum decrease in eigenvalue. We find that GreedyWalk performs better than all the heuristics in all the networks we study. In the experiments, we analyze GreedyWalk for walks of length $k = (\log n)$.

Organization. The background and notation are defined in Section 2. Sections 3, 4 and 5 cover GreedyWalk, GreedyWalkSparse and PrimalDual algorithms, respectively, for the SMRE problem; the SRMN problem is discussed in section 6. Some of the algorithmic details and proofs are omitted for brevity and are available in [34]. Lower bounds for some heuristics and the experimental results are discussed in Sections 7 and 8, respectively. We discuss the related work in Section 9 and conclude in Section 10.

2 Preliminaries

Table 1: Notations

$G = (V, E)$	Graph representing a contact network
$n = V $	Total number of nodes in G
$d(v, G)$	Degree of node v in G
$\Delta(G)$	Maximum node degree in G
$A = A^G$	Adjacency matrix of G
$G[E]$	Subgraph of G induced on E
$\lambda_i(G)$	i th largest Eigenvalue of A^G
$\lambda_1(G) = \rho(A)$	$\lambda_1(G)$, spectral radius of G
$c(v)$	Cost of a vertex or edge of G
β	Infection rate
γ	Recovery rate
T	Epidemic Threshold, $T = \gamma/\beta$
	Time to epidemic extinction
$W_k(G)$	Set of closed walks of length k in G
$ W_k(G) $	$ W_k(G) = W_k(G) $
$\text{nodes}(w)$	number of distinct nodes in walk w
$\text{walks}(x, G, k)$	Number of closed k -walks in G containing edge (or vertex) x
$E_{\text{OPT}}(T)$	Optimal solution to SRME($G, c(\cdot), T$)

We consider undirected graphs $G = (V, E)$, and interventions to control the spread of epidemics via vaccination (modeled by removal of nodes) and quarantining (modeled by removal of edges). There can be different

costs for the removal of nodes and edges (denoted by $c(v)$ and $c(e)$, respectively), e.g., depending on their demographics, as estimated by [25]. For a set $E \subseteq E$, $c(E) = \sum_{e \in E} c(e)$ denotes the total cost of the set E (similarly for node subsets).

There are a number of models for epidemic spread; we focus on the fundamental SIS (Susceptible-Infectious-Susceptible) model, which is defined in the following manner. Nodes are in susceptible (S) or infectious (I) state. Each infected node u (in state I) causes each susceptible neighbor v (in state S) to become infected at rate β_{uv} . Further, each infected node u switches to the susceptible state at rate γ . In this paper, we assume a uniform rate $\beta_{uv} = \beta$ for all $(u, v) \in E$; in this case, we define a threshold $T = \gamma/\beta$, which characterizes the time to extinction. Let $A = A^G$ denote the adjacency matrix of G , and let $n = |V|$. Let $\lambda_i(G)$ denote the i th largest eigenvalue of A , and let $\rho(A) = \lambda_1(A)$ denote the spectral radius of A . Since G is undirected, it follows that all eigenvalues are real, and $\rho(A) > 0$ (see, e.g., Chapter 3 of [26]). Ganesh et al. [13] showed that the epidemic dies out in time $O(\frac{\log n}{1 - \rho(A)/T})$, if $\rho(A) < T$ in the SIS model, with high probability; this threshold was also observed by [39]. Prakash et al. [30] showed this condition holds for a broad class of other epidemic models, including the SIR model (which contains the Recovered state). Now we formally define the SRM problem.

Definition 2.1. Spectral Radius Minimization problems (SRME and SRMN): Given an undirected graph $G = (V, E)$, with cost $c(e)$ for each edge e , and a threshold T , the goal of the SRME($G, c(\cdot), T$) problem is to find the cheapest subset $E \subseteq E$ such that $\lambda_1(G[E]) < T$. We refer to the node version of this problem as SRMN($G, c(\cdot), T$).

We discuss some notation that will be used in the rest of the paper. $E_{\text{OPT}}(T)$ denotes an optimal solution to the SRME($G, c(\cdot), T$) problem. Let $W_k(G)$ denote the set of closed walks of length k in G ; let $|W_k(G)| = |W_k(G)|$. For a walk w , let $\text{nodes}(w)$ denote the number of distinct nodes in w . A standard result (see, e.g., Chapter 3 of [26]) is the following:

$$(2.1) \quad \text{nodes}(w) = \frac{1}{|W_k(G)|} \sum_{i=1}^n A_{ii}^k = \sum_{i=1}^n \lambda_i(G)^k.$$

The number of walks in $W_k(G)$ containing a node i is A_{ii}^k . For a graph G , let $\text{walks}(e, G, k)$ denote the number of closed k -walks in G containing $e = (i, j)$. Then, $\text{walks}(e, G, k) = A_{ij}^{k-1}$. We say that an edge set E hits a walk w if w contains an edge from E . Similarly, for a node v , let $\text{walks}(v, G, k)$ denote the number of closed

k -walks in G containing v . Then, $\text{walks}(i, G, k) = A_{ii}^k$. Table 1 summarizes the frequently used notations.

3 GreedyWalk: $O(\log n \log \frac{1}{\epsilon})$ -approximation

Main idea. Our starting point is the connection between the number of closed walks in a graph and the sum of powers of the eigenvalues in (2.1). We try to reduce the spectral radius by reducing the number of closed walks of length k in the graph, by removing edges (see Algorithm 1). This, in turn, can be viewed as a partial covering problem.¹ Our basic idea extends to the node version as well, as discussed later in Section 6.

Algorithm 1 GreedyWalk (high level description)

Input: $G, T, c(E) \geq k$ even

Output: Edge set E

```

1: Initialize  $E$ 
2: while  $W_k(G[E \setminus E]) \geq nT^k$  do
3:    $r = \min_{e \in E} \frac{\text{walks}(e, G[E \setminus E], k)}{c(e)}$ 
4:   Pick  $e \in E$  that maximizes  $\frac{\text{walks}(e, G[E \setminus E], k)}{c(e)}$ 
5:    $E = E \setminus \{e\}$ 
6: end while

```

The Lemma below proves the approximation bound for any solution (say E) from GreedyWalk. Let $G = G[E \setminus E]$ denote the graph resulting after the removal of edges in E . Our proof involves three steps: (1) Proving the bound on $\sum_{i=1}^n \lambda_i(G)^k$; (2) Relating $c(E)$ to the cost of the optimum solution to the partial covering problem which ensures that the number of walks in the residual graph is at most nT^k ; (3) Showing that the optimum solution to the SRME problem also ensures that at most nT^k remain in the residual graph.

Lemma 3.1. *Let E denote the set of edges found by Algorithm GreedyWalk. Given any constant $\epsilon > 0$, let k be an even integer larger than $\frac{\log n}{\log(1+\epsilon)}$. Then, we have $\sum_{i=1}^n \lambda_i(G[E \setminus E])^k \leq (1+\epsilon)T$, and $c(E) = O(c(E_{\text{OPT}}(T)) \log n \log \frac{1}{\epsilon})$.*

Proof. We follow the proof scheme mentioned above. By the stopping condition of the algorithm, we have $W_k(G) \geq nT^k$. From (2.1), we have $\sum_{i=1}^n \lambda_i(G)^k = \sum_{i=1}^n A_{ii}^k = \sum_{w \in W_k(G)} \text{nodes}(w) = kW_k(G)$, which implies $\sum_{i=1}^n \lambda_i(G)^k \geq nkT^k$. Further, since k is even

(by assumption), $\lambda_i(G) \geq 0$, so that $\sum_{i=1}^n \lambda_i(G)^k \leq \sum_{i=1}^n \lambda_i(G)^k \leq nkT^k$. This implies $\sum_{i=1}^n \lambda_i(G)^k \leq e^{(\log n + \log k)/k} T$. Since $k = \log n / \log(1+\epsilon)$, we have $(\log n + \log k)/k \leq 2 \log(1+\epsilon)$, so that $\sum_{i=1}^n \lambda_i(G)^k \leq (1+\epsilon)^2 T \leq (1+\epsilon)T$.

Next, we derive a bound for $c(E)$. Observe that the algorithm can be viewed as solving a partial cover problem, in which (i) the set H of elements corresponds to walks in $W_k(G)$, and (ii) there is a set corresponding to each edge $e \in E$ consisting of all the walks in $W_k(G)$ that contain e . Following the analysis of the greedy algorithm for partial cover [35], we have $c(E) = O(c(E_{\text{HITOPT}}) \log |H|)$, where E_{HITOPT} denotes the optimum solution for this covering instance. Since Δ denotes the maximum node degree, we have $|H| = W_k(G) \leq n^k$. We show below that $c(E_{\text{HITOPT}}) \leq c(E_{\text{OPT}}(T))$; it follows that $c(E) = O(c(E_{\text{OPT}}(T)) \log n \log \frac{1}{\epsilon})$.

Finally, we prove that $c(E_{\text{HITOPT}}) \leq c(E_{\text{OPT}}(T))$. By definition of $E_{\text{OPT}}(T)$, we have $\sum_{i=1}^n \lambda_i(G[E \setminus E_{\text{OPT}}(T)]^k \leq T$. Let $G = G[E \setminus E_{\text{OPT}}(T)]$. Then, we have

$$W_k(G) = \sum_{i=1}^n \lambda_i(G)^k \leq n \sum_{i=1}^n \lambda_i(G)^k \leq nT^k.$$

This implies $E_{\text{OPT}}(T)$ hits at least $W_k(G) \geq nT^k$ walks, so that $c(E_{\text{HITOPT}}) \leq c(E_{\text{OPT}}(T))$.

Effect of the walk length k . We set the walk length $k = a \log n$ for some constant a in Algorithm GreedyWalk; understanding the effect of k is a natural question. From the proof of Lemma 3.1, it follows that $\sum_{i=1}^n \lambda_i(G[E \setminus E])^k$ can be bounded by $(nk)^{1/k} T$ for any choice of k , as long as it is even. This bound becomes worse as k becomes smaller, e.g., it is $O(\sqrt{n})$ for $k = 2$.

In order to complete the description of GreedyWalk (Algorithm 1), we need to design an efficient method to determine the edge which maximizes the quantity in line 4. We discuss two methods below.

3.1 Matrix multiplication approach for implementing GreedyWalk. Note that $\text{walks}(e, G, k) = A_e^k$. We use matrix multiplication to compute A_e^k once for each iteration of the while loop in line 2 of Algorithm 1. In line 4, we iterate over all edges, in order to compute the edge e that maximizes the given ratio. For $k = O(\log n)$, A_e^k can be computed in time $O(n \log \log n)$, where $\epsilon < 2.37$ is the exponent for the running time of the best matrix multiplication algorithm [40]. Therefore, each iteration involves $O(n \log \log n + m) = O(n \log \log n)$ time. This gives a total running time of $O(n \log \log n / E_{\text{OPT}} / \log^2 n)$, since only $O(E_{\text{OPT}} / \log^2 n)$ edges are removed. One draw-

¹This is a variation of the set cover problem, in which an instance consists of (i) a set H of elements, (ii) a collection $S = \{S_1, \dots, S_m\} \subseteq 2^H$ of sets, (iii) $\text{cost}(S_i)$ for each $S_i \in S$, and (iv) a parameter $r = |H|$. The objective is to find the cheapest collection of sets from S which cover at least r elements. Slavic [35] shows that a greedy algorithm gives an $O(\log |H|)$ approximation.

back with this approach is the high (super-linear) space complexity, even with the best matrix multiplication methods, in general.

3.2 Dynamic programming approach for implementing GreedyWalk. When the graphs are very sparse ($\epsilon(n)$ edges), we adapt a dynamic programming approach to compute $\text{walks}(e, G, k)$ for an edge e and more efficiently select the edge that maximizes $\text{walks}(e, G[E \setminus E], k)/c(e)$ in line 4 of Algorithm 1. Although, potentially $\text{walks}(e, G, k)$ needs to be computed for each edge $e \in E \setminus E$, in practice it suffices to compute it for only a small subset of $E \setminus E$. We make use of the fact that $\text{walks}(e, G, k) \geq \text{walks}(e, G, k)$ for any subgraph G . The approach is briefly as follows. Initially we compute $\text{walks}(e, G, k)$ for each $e \in E$ and arrange the edges in non-ascending order of their $\text{walks}(e, G, k)$ value, $e_1, e_2, \dots, e_{|E|}$. After the i -th edge (i.e., e_i in the i -th iteration) is removed, $\text{walks}(e, G, k)$ is computed on the residual graph G only for some consecutive edges in that order upto some e_j such that $\text{walks}(e_i, G, k) > \text{walks}(e_{j+1}, G, k)$. Edges e_2, \dots, e_j are reordered based on the recomputed walk numbers, $\text{walks}(e_i, G, k)$ and then the same steps are repeated. The approach takes $O(n)$ space and $O(n^2 k)$ time assuming the number of edges is $\epsilon(n)$ in real world large networks. The algorithmic details and proofs are omitted for brevity.

4 Using sparsification for faster running time: Algorithm GreedyWalkSparse

The efficiency of Algorithm GreedyWalk can be improved if the number of edges in the graph can be reduced. This can be achieved by two pruning steps - pruning edges such that in the residual graph (i) no node has degree more than T^2 , and (ii) there is no T -core; the T -core of a graph denotes the maximal subgraph of G with minimum degree T (see, e.g., [1]). We will refer to these steps as MaxDegreeReduction and DensityReduction respectively. This leads to sparser graphs, without affecting the asymptotic approximation guarantees.

Lemma 4.1. *Let E_1 and E_2 denote the set of edges removed in the pruning steps MaxDegreeReduction and DensityReduction, respectively. Then, $c(E_1)$ and $c(E_2)$ are both at most $2c(E_{\text{OPT}}(T))$.*

The proof of the lemma and algorithmic details of sparsification are omitted for brevity. By Lemma 4.1, it follows that the approximation bounds of Lemma 3.1 still hold. However, the pruning steps reduce the number of edges, thereby speeding the implementation of GreedyWalk. We discuss the empirical performance of pruning in Section 8. We show below that pruning

also improves the approximation factor marginally from $O(\log n \log T)$ to $O(\log n \log T)$ which could be significant when n is large and T is small.

Lemma 4.2. *Let E denote the set of edges found by Algorithm GreedyWalkSparse. Given any constant $\epsilon > 0$, let k be an even integer larger than $\frac{\log n}{\log(1+\epsilon/3)}$. Then, we have $|E| \leq (1 + \epsilon)T$, and $c(E) = O(c(E_{\text{OPT}}(T)) \log n \log T)$.*

The proof is omitted for brevity.

5 Primal Dual: $O(\log n)$ -approximation

Main idea: The approach of [12] gives an f -approximation for the partial covering problem, where f denotes the maximum number of sets that contain any element in the set system. As in the proof of Lemma 3.1, in our reduction from the SRME problem to partial covering, elements correspond to all the closed walks of length $k = O(\log n)$, while sets correspond to edges; for an edge e , the corresponding set S_e consists of all the walks w that are hit by e . In this reduction, each walk w lies in k sets; therefore, $f = O(\log n)$ for this set system. Therefore, the approach of [12] could improve the approximation factor. Unfortunately, our set system has size $n^{O(\log n)}$, so that the algorithm of [12] cannot be used directly to get a polynomial time algorithm.

Algorithm 2 Primal Dual (T, S, c, ϵ)

Output: Edge set E

```

1: Initialize  $z_e = 0$  for all  $S_e \in S, C = \emptyset$ .
2:  $// u(w) = 0$  for all walks  $w$  in  $G$ .
3: while  $C$  is not  $\epsilon$ -feasible do
4:    $x = \min_{e \in E \setminus E} \{ \frac{c(e)z_e}{\text{walks}(e, G, k)} \}$ ; let  $e$  be an edge for
     which the minimum is reached.
5:    $C = C \cup \{S_e\}$ 
6:   For each  $e \in E \setminus E : z_e = z_e + x \cdot \text{walks}(e, G, k)$ 
7:    $// u(w) = u(w) + x$  for all walks  $w$  in  $G$  that
     pass through  $e$ 
8:    $E = E \cup \{e\}$ 
9: end while
```

The algorithm of Gandhi et al. [12] uses a primal-dual approach, which maintains dual variables $u(w)$ for each element (i.e., walk); these are increased gradually, and a set (i.e., an edge) is picked if the sum of duals corresponding to the elements in the set equals its cost. We now discuss how to adapt this algorithm to run in polynomial time, and only focus on polynomial time implementation of the Primal Dual subroutine of [12] in detail here. However, we also present the set cover algorithm HitWalks for completeness. This algorithm iterates over all edges and invokes Primal Dual in each iteration to obtain a candidate set of edges to remove

and \mathcal{W} chooses the set with minimum cost. T, S, c and \mathcal{W} denote the set of elements (walks) to be covered, the sets (corresponding to edges that can be chosen), the costs corresponding to the sets/edges and the number of elements (walks) that need to be covered, respectively. A subset $C \subseteq S$ is \mathcal{W} -feasible if $\bigcup_{e \in C} S_e = \mathcal{W}$. Let $u(w)$ denote the dual variables corresponding to the walks w ; these are not maintained in the algorithm explicitly, but assigned in the comments, for use in the analysis. This algorithm does not explicitly update the

Algorithm 3 Hit Walks(T, S, c, \mathcal{W})

Input: Set of all k -closed walks T , walks corresponding to edges S , edge cost set c , number of walks to hit

Output: Edge set E

```

1: Sort the edges of  $G$  in increasing order of their costs.
2: Initialize  $j, c(e_j)$ 
3: for  $j = 1$  to  $m$  do
4:    $c(e_j) = c(e_j)$  and compute  $\text{walks}(e_j, G, k)$ 
5:    $cs_j = \dots$  //cost of edge set in this iteration
6:   if  $|S_1 \cup S_2 \cup \dots \cup S_j| < |\mathcal{W}|$  then
7:      $E_j = \{e_j\}$  Primal Dual  $T \setminus S_j, S \setminus S_j, c, \mathcal{W}$ 
        $\text{walks}(e_j, G, k)$ 
8:      $cs_j = c(E_j)$ 
9:   end if
10:   $i = \min_j cs_j$ 
11:   $E = E_i$ 
12: end for
```

dual variables, but the edges are picked in the same sequence as in [12].

Lemma 5.1. *Given any constant $\epsilon > 0$, let k be an even integer larger than $\frac{\log n}{\log(1+\epsilon)}$. The dual variables $u(w)$ in algorithm Primal Dual are maintained and updated as in [12], and the edge e picked in each iteration is the same. We have $c(E) = O(c(E_{\text{OPT}}) \log n)$ and $1(G[E \setminus E]) \leq (1 + \epsilon)T$.*

The proof is omitted for brevity.

6 Node Version

Our discussion so far has focused on the SRME problem. We now consider The SRMN problem. Recall the definition of $\text{walks}(v, G, k)$ from Section 2. Let $G[V \setminus V]$ denote the subgraph of $G = (V, E)$ induced by subset $V \subseteq V$. We modify Algorithm GreedyWalk to work for the SRMN problem as shown in algorithm GreedyWalkSRMN.

It can be shown on the same lines as Lemma 3.1 that this gives a solution of cost $O(c(E_{\text{OPT}}(T)) \log n \log \dots)$, where $c(E_{\text{OPT}}(T))$ denotes the cost of the optimal solution to SRMN problem. Further, the same running time bounds as in Sections 3.1 and 3.2 hold.

Algorithm 4 Algorithm GreedyWalkSRMN

```

1: Initialize  $V$ 
2: while  $W_k(G[V \setminus V]) \neq nT^k$  do
3:    $r = W_k(G[E \setminus E]) \leq nT^k$ 
4:   Pick  $v \in V \setminus V$  that maximizes
        $\frac{\min\{r, \text{walks}(v, G[V \setminus V], k)\}}{c(v)}$ 
5:    $V = V \cup \{v\}$ 
6: end while
```

7 Popular heuristics and lower bounds

A number of heuristics have been developed for controlling the spread of epidemics; these are discussed below. All these heuristics involve ordering the edges based on some kind of score, and then selecting the top few edges based on this score. We describe the score function in each heuristic.

1. **ProductDegree** ([27]): The score for edge $e = (u, v)$ is defined as $\deg(u) \deg(v)$. Edges are removed in non-increasing order of this score.
2. **EigenScore** ([27, 37]): Let \mathbf{x} be the eigenvector corresponding to the λ_1 eigenvalue of the graph. The score for edge $e = (u, v)$ is $|x(u) - x(v)|$.
3. **LinePagerank**: This method uses the linegraph $L(G) = (E, F)$ of graph $G = (V, E)$, where (e, e') $\in F$ if $e, e' \in E$ have a common endpoint. We define the score of edge $e \in E$ as the pagerank of the corresponding node in $L(G)$.

As we said in Section 8.2, these heuristics work well for different kinds of networks. We design another heuristic, Hybrid, which picks the best of the EigenScore and ProductDegree methods. The edges are ordered in the following manner: (1) Let i_1, \dots, i_m and j_1, \dots, j_m be orderings of edges in the Eigenscore and ProductDegree algorithms, respectively. (2) Initialize $i = 0$ and $j = 0$, and (3) from the edges (i) and (j) , remove the one which decreases the max eigenvalue of the residual graph more. Increment the corresponding index.

We have examined the worst case performance of these heuristics. Two of these, namely, EigenScore and ProductDegree, have been used specifically for reducing the spectral radius, e.g., [27, 37]. No formal analysis is known for any of these heuristics in the context of the SRME or SRMN problems; some of them seem to work pretty well on real world networks. We show that the worst case performance of these heuristics can be quite poor, in general.

Theorem 7.1. *Given any sufficiently large positive integer n , there exists a threshold $T < a \bar{n}$, for*

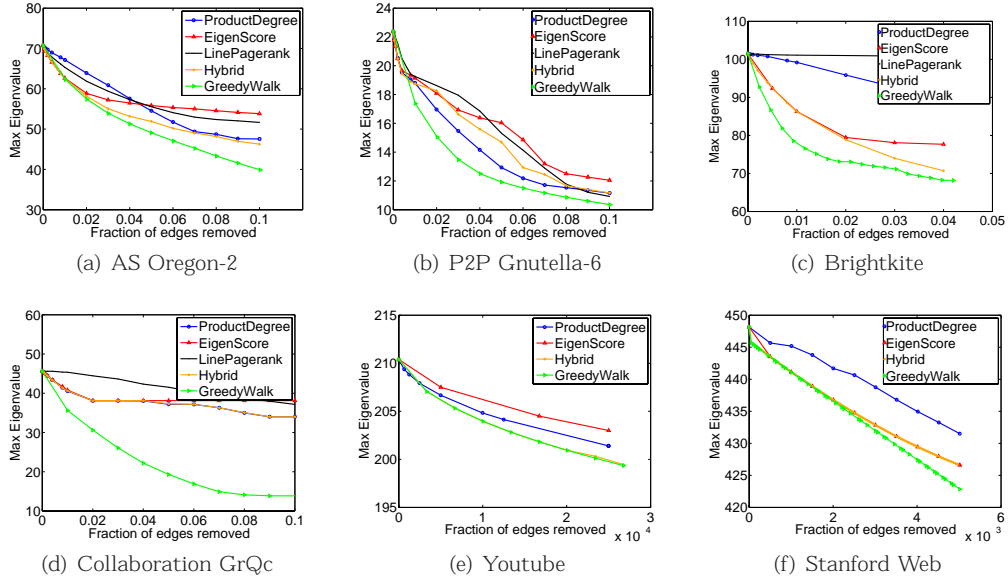


Figure 1: Comparison between the GreedyWalk, ProductDegree, Eigenscore, LinePagerank and Hybrid algorithms for different networks. Each plot shows the spectral radius (y-axis) as a function of the fraction of edges removed (x-axis). The LinePagerank heuristic has not been evaluated in 1e and 1f because of the scale of these networks.

some constant $a < 1$ and a graph of size n for which the number of edges removed by ProductDegree, Eigenscore, Hybrid and LinePagerank is $\frac{n}{T^2} c(E_{OPT})$.

We give here a proof sketch for theorem 7.1. The detailed proof is omitted for brevity. We construct a graph G for which the statement holds. For convenience let us assume that T is a positive integer. G contains (1) a clique G_1 on $T + 1$ nodes; (2) a caterpillar tree G_2 , which comprises of a path $v_1 v_2 \dots v_q$ with v_i adjacent to T leaves each and (3) G_3 , a star graph with $(T + 1)^2$ leaves and central vertex denoted by v_q . We connect G_1 to G_2 by (v_0, v_1) where, v_0 is some node in G_1 and G_2 is connected to G_3 by the edge (v_q, v_{q+1}) . Note that $q = \frac{n - (T + 1)^2}{T}$ and $\lambda_1(G) = \lambda_1(G_3) = T + 1$. Again, here we assume that q is an integer.

It can be shown that $c(E_{OPT}) = 2T + 3$. Removing the edges (v_0, v_1) and (v_q, v_{q+1}) isolates the components G_1 , G_2 and G_3 . G_1 is a clique on $T + 1$ nodes and on removing one edge, its spectral radius decreases below T . G_2 is a star with $(T + 1)^2$ leaves and therefore, on removing at most $(T + 1)^2$ edges, its spectral radius decreases below T . It can be shown that $\lambda_1(G_2) = T + 2$.

It can be demonstrated that all the four algorithms score the edges (v_i, v_{i+1}) , $i = 0, \dots, q$ above any edge belonging to the clique G_1 . However, the spectral

radius cannot be brought down below T until at least one edge in G_1 is removed. Therefore, at least q edges will be removed by all the algorithms. By the initial assumption that $T < c \frac{n}{T}$, it follows that $q = \frac{n}{T}$, while, $c(E_{OPT}) = O(T)$, hence the theorem holds.

8 Experiments

8.1 Methods and Dataset We evaluate the algorithms developed in the paper² GreedyWalk, GreedyWalkSparse and PrimalDual and compare their performance with the heuristics from literature Eigenscore, ProductDegree, LinePagerank and Hybrid (described in Section 7), as a more sophisticated baseline. The networks which we have considered in our empirical analysis are listed in Table 2 spanning infrastructure networks, social networks and random graphs.

8.2 Experimental results

Performance of our algorithms and comparison with other heuristics: We compare the quality of solutions from our algorithms with the Eigenscore, ProductDegree, LinePagerank and Hybrid heuristics in Figure 1. We note that GreedyWalk is consistently better than all other heuristics, especially as the target threshold becomes smaller.

²All code at: <http://tinyurl.com/l3lgsq7>.

Table 2: Networks and their sizes. The first two are synthetic random networks; others are taken from [36] and [10]

Network	nodes	edges	λ_1
Barabasi-Albert	1000	1996	11.1
Erdos-Renyi	994	2526	6.38
P2P (Gnutella05)	8846	31839	23.55
P2P (Gnutella06)	8717	31525	22.38
Collab. Net (HepTh)	9877	25998	31.03
Collab. Net (GrQc)	5242	14496	45.62
AS (Oregon 1)	10670	22002	58.72
AS (Oregon 2)	10900	31180	70.74
Brightkite Net	58228	214078	101.49
Youtube Network	1134890	2987624	210.4
Stanford Web graph	281903	1992636	448.13

Compared to the EigenScore, Product Degree and LinePagerank heuristics, the spectral radius for the solution produced by GreedyWalk, as a function of the fraction of edges removed, is lower by at least 10-20%. Our improved baseline, the Hybrid heuristic, works better than the other heuristics, and comes somewhat close to GreedyWalk in many networks.

Though PrimalDual gives a significantly better approximation guarantee, compared to GreedyWalk, it has a much higher running time. Therefore, we only evaluate it for one iteration of Algorithm HitWalks. Figure 2 shows that PrimalDual is quite close to GreedyWalk after just one iteration; we expect running this algorithm fully would further improve the performance, but additional work is needed to improve the running time.

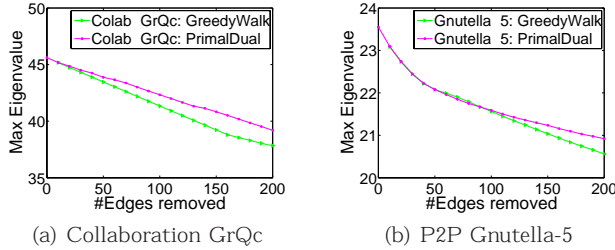


Figure 2: GreedyWalk vs PrimalDual. Each plot shows the spectral radius (y-axis) as a function of the number of edges removed (x-axis) using the two methods.

Running time and effect of sparsification: Figure 3 shows the total running time of GreedyWalk for three networks. The time decreases with the increase of T , because the while loop in Algorithm GreedyWalk needs to be run for fewer iterations. The high running time motivates faster methods. We evaluate the performance of the GreedyWalkSparse algorithm. As

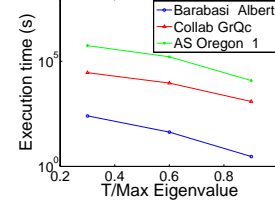


Figure 3: Total running time of GreedyWalk method (y-axis) as a function of $T/\lambda_1(G)$ (x-axis), where T is the threshold and $\lambda_1(G)$ is the spectral radius of the initial graph, without any edges removed.

shown in Figure 4, GreedyWalkSparse gives almost the same quality of approximation as GreedyWalk, but improves the running time by up to an order of magnitude, particularly when T is small.

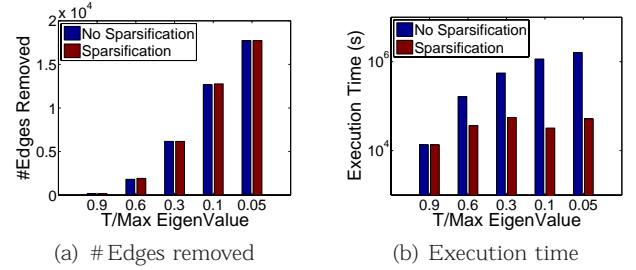


Figure 4: Impact of sparsification on GreedyWalk. The plots show for AS Oregon-1 network, (a) the number of edges removed and (b) the execution time on the y-axis, as a function of $T/\lambda_1(G)$ (x-axis), where T is the threshold and $\lambda_1(G)$ is the spectral radius of the initial graph, without any edges removed.

SRMN: For the SRMN problem, we compare the adaption of GreedyWalk, as discussed in Section 6, with the node versions of the Degree and EigenScore heuristic [37]. As shown in Figure 5, GreedyWalk performs consistently better.

Main observations:

1. GreedyWalk performs consistently better than existing heuristics in removing nodes or edges.
2. Performance of just one iteration of PrimalDual comes very close to GreedyWalk.
3. Sparsification helps in improving the speed of GreedyWalk without affecting the solution quality.

9 Related Work

Related work comes from multiple areas: epidemiology, immunization algorithms and other optimization algorithms. There is general research interest in studying

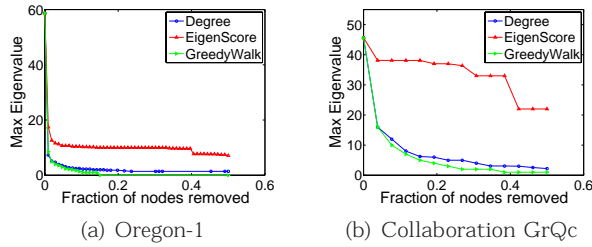


Figure 5: Computing solutions for SRMN problem on Oregon-1 and collaboration GrQc network with GreedyWalk algorithm, adapted as in Section 6, and Degree and EigenScore heuristics [37]. The plots show the resultant spectral radius (y-axis) as fractions of nodes are removed (x-axis) with different methods.

dynamic processes on large graphs, (a) blogs and propagations [16, 21], (b) information cascades [14, 15] and (c) marketing and product penetration [33]. These dynamic processes are all closely related to virus propagation.

Epidemiology: A classical text on epidemic models and analysis is by May and Anderson [2]. Most work in epidemiology is focused on *homogeneous models* [4, 2]. Here we study network based models. Much work has gone into in finding epidemic thresholds (minimum virulence of a virus which results in an epidemic) for a variety of networks [28, 39, 13, 30].

Immunization: There has been much work on finding optimal strategies for vaccine allocation [5, 24, 9]. Cohen et al [11] studied the popular *acquaintance* immunization policy (pick a random person, and immunize one of its neighbors at random). Using game theory, Aspnes et al. [3] developed inoculation strategies for victims of viruses under random starting points. Kuhlman et al. [20] studied two formulations of the problem of blocking a contagion through edge removals under the model of discrete dynamical systems. As already mentioned Tong et al. [38, 37], Van Miegham et al. [27], Prakash et al. [29] and Chakrabarti et al. [7] 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. [6], He et al. [17] and Khalil et al. [19].

Other Optimization Problems: Other diffusion based optimization problems include the influence maximization problem, which was introduced by Domingos and Richardson [32], and formulated by Kempe et al. [18] 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. Other such problems where we wish to select a subset of *important* vertices on graphs, include outbreak detection [23] and finding most-likely culprits of epidemics [32, 31].

10 Conclusions

We study the problem of reducing the spectral radius of a graph to control the spread of epidemics by removing edges (the SRME problem) or nodes (the SRMN problem). We have developed a suite of algorithms for these problems, which give the *best* rigorous bounds for these problems. Our main algorithm GreedyWalk performs consistently better than all other heuristics for these problems, in all networks we studied. We also develop variants that improve the running time by sparsification, and improve the approximation guarantee using a primal dual approach. These algorithms exploit the connection between the graph spectrum and closed walks in the graph, and perform better than all other heuristics. Improving the running time of these algorithms is a direction for further research. We expect these techniques could potentially help in optimizing other objectives related to spectral properties, e.g., *robustness* [8], and in other problems related to the design of interventions to control the spread of epidemics.

Acknowledgments. 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 ICES CCF-1216000, NSF NETSE Grant CNS-1011769, DOE DE-SC0003957, NSF Grant IIS-1353346 and Maryland Procurement Office contract H98230-14-C0127. 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.

Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/NBC, or the US Government.

References

- [1] A. Adiga and A. Vullikanti. How robust is the core of a network? In *Proc. of ECMLPKDD*, 2013.
- [2] R. M. Anderson and R. M. May. *Infectious Diseases of Humans*. Oxford University Press, 1991.
- [3] J. Aspnes, K. Chang, and A. Yampolskiy. Inoculation strategies for victims of viruses and the sum-of-squares partition problem. In *Proc. of ACM SODA*, 2005.

- [4] N. Bailey. *The Mathematical Theory of Infectious Diseases and its Applications*. Griffin, London, 1975.
- [5] L. Briesemeister, P. Lincoln, and P. Porras. Epidemic profiles and defense of scale-free networks. *WORM*, Oct 2003.
- [6] C. Budak, D. Agrawal, and A. E. Abbadi. Limiting the spread of misinformation in social networks. In *Proc. of WWW*, 2011.
- [7] D. Chakrabarti, Y. Wang, C. Wang, J. Leskovic, and C. Faloutsos. Epidemic thresholds in real networks. *ACM Transactions on Information and System Security (TISSEC)*, 2008.
- [8] H. Chan, H. Tong, and L. Akoglu. Make it or break it: Manipulating robustness in large networks. In *Proc. of SDM*, pages 325–333, 2014.
- [9] P. Chen, M. David, and D. Kempe. Better vaccination strategies for better people. In *Proc. of ACM conference on Electronic commerce (EC)*. ACM, 2010.
- [10] CINET: Cyber infrastructure for network science. http://cinet.vbi.vt.edu/cinet_new/.
- [11] R. Cohen, S. Havlin, and D. ben Avraham. Efficient immunization strategies for computer networks and populations. *Physical Review Letters*, 91(24), 2003.
- [12] R. Gandhi, S. Khuller, and A. Srinivasan. Approximation algorithms for partial covering problems. *Journal of Algorithms*, 53(1):55–84, 2004.
- [13] A. Ganesh, L. Massoulié, and D. Towsley. The effect of network topology on the spread of epidemics. In *Proc. of INFOCOM*, 2005.
- [14] 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.
- [15] M. Granovetter. Threshold models of collective behavior. *Am. Journal of Sociology*, 83(6):1420–1443, 1978.
- [16] D. Gruhl, R. Guha, D. Liben-Nowell, and A. Tomkins. Information diffusion through blogspace. In *Proc. of WWW*, 2004.
- [17] X. He, G. Song, W. Chen, and Q. Jian. Influence blocking maximization in social networks under the competitive linear threshold model. In *Proc. of SDM*, 2012.
- [18] D. Kempe, J. Kleinberg, and A. Tardos. Maximizing the spread of influence through a social network. In *Proc. of KDD*, New York, NY, 2003. ACM Press.
- [19] E. Khalil, B. Dilikina, and L. Song. Scalable diffusion-aware optimization of network topology. In *Proc. of KDD*, 2014.
- [20] C. J. Kuhlman, G. Tuli, S. Swarup, M. V. Marathe, and S. S. Ravi. Blocking simple and complex contagion by edge removal. In *Proc. of ICDM*, pages 399–408, 2013.
- [21] R. Kumar, J. Novak, P. Raghavan, and A. Tomkins. On the bursty evolution of blogspace. In *Proc. of WWW*, 2003.
- [22] T. Lappas, E. Terzi, D. Gunopulos, and H. Mannila. Finding e-ectors in social networks. In *Proc. of SIGKDD*, pages 1059–1068, 2010.
- [23] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. S. Glance. Cost-effective outbreak detection in networks. In *Proc. of KDD*, 2007.
- [24] N. Madar, T. Kalisky, R. Cohen, D. ben Avraham, and S. Havlin. Immunization and epidemic dynamics in complex networks. *Eur. Phys. J. B*, 38(2):269–276, 2004.
- [25] J. Medlock and A. P. Galvani. Optimizing influenza vaccine distribution. *Science*, 325(5948), 2009.
- [26] P. V. Mieghem. *Spectral Graph Theory*. Cambridge University Press, 2011.
- [27] 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.
- [28] R. Pastor-Satorras and A. Vespignani. Epidemic dynamics in finite size scale-free networks. *Physical Review E*, 65:035108, 2002.
- [29] B. A. Prakash, L. A. Adamic, T. J. Iwashyna, H. Tong, and C. Faloutsos. Fractional immunization in networks. In *Proc. of SDM*, pages 659–667, 2013.
- [30] 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.
- [31] B. A. Prakash, J. Vreeken, and C. Faloutsos. Spotting culprits in epidemics: How many and which ones? In *Proc. of ICDM*, 2012.
- [32] M. Richardson and P. Domingos. Mining knowledge-sharing sites for viral marketing. In *Proc. of KDD*, 2002.
- [33] E. M. Rogers. *Diffusion of Innovations*, 5th Edition. Free Press, August 2003.
- [34] S. Saha, A. Adiga, B. A. Prakash, and A. Vullikanti. Reducing the spectral radius to control epidemic spread. Technical report, available at <http://stanford.vbi.vt.edu/ssaha/papers/eigext.pdf>.
- [35] P. Slavik. Improved performance of the greedy algorithm for partial cover. *Information Processing Letters*, 1997.
- [36] SNAP: Stanford network analysis project. <http://snap.stanford.edu/index.html>.
- [37] 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.
- [38] 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.
- [39] Y. Wang, D. Chakrabarti, C. Wang, and C. Faloutsos. Epidemic spreading in real networks: An eigenvalue viewpoint. In *Proc. of SRDS*, pages 25–34, Los Alamitos, CA, 2003. IEEE Computer Society Press.
- [40] V. V. Williams. Multiplying matrices faster than coppersmith-winograd. In *Proc. of STOC*, 2012.