Fractional Immunization in Networks

B. Aditya Prakash* Lada Adamic† Theodore Iwashyna‡ Hanghang Tong§
Christos Faloutsos¶

Abstract

Preventing contagion in networks is an important problem in public health and other domains. Targeting nodes to immunize based on their network interactions has been shown to be far more effective at stemming infection spread than immunizing random subsets of nodes. However, the assumption that selected nodes can be rendered completely immune does not hold for infections for which there is no vaccination or effective treatment. Instead, one can confer fractional immunity to some nodes by allocating variable amounts of infection-prevention resource to them. We formulate the problem to distribute a fixed amount of resource across nodes in a network such that the infection rate is minimized, prove that it is NP-complete and derive a highly effective and efficient linear-time algorithm. We demonstrate the efficiency and accuracy of our algorithm compared to several other methods using simulation on real-world network datasets including US-MEDICARE and state-level interhospital patient transfer data. We find that concentrating resources at a small subset of nodes using our algorithm is up to *6 times* more effective than distributing them uniformly (as is current practice) or using network-based heuristics. To the best of our knowledge, we are the *first* to formulate the problem, use truly nation-scale network data and propose effective algorithms.

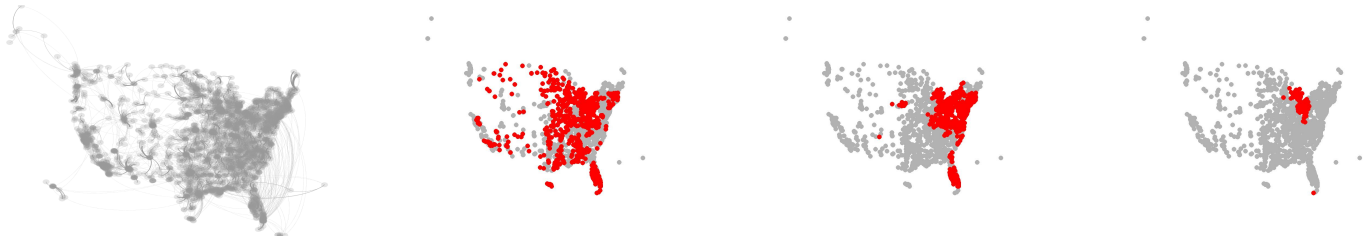
1 Introduction

Given a graph and vaccines which provide partial ('fractional') protection, how to distribute them to maximize lives saved? Networks carry harmful agents, e.g. disease, computer viruses, and even misinformation. The networks' structure dictates how rapidly the malicious agent will spread. One can take advantage of this structure to identify specific nodes for infection control, such that the spread of the disease is significantly curtailed. In selecting nodes for infection control, previous work has assumed that nodes can be rendered completely immune. However, in many cases the complete immunization of a node is not an option. Bacteria present in hospitals have developed resistance to most antibiotics. Vaccines take time to be developed for both human and computer viruses, prompting other measures to prevent epidemics. However, one can provide partial ('fractional') protection by allocating resources to render nodes less susceptible.

In this paper we formulate the problem of distributing resources to minimize the spread of infection on a network. Previously devised models, which assume that allocating a single unit of resource to a node renders it completely immune are a special case of this more general problem. We illustrate the problem in two settings: the spread of infection between hospitals through patient transfers, and the spread of malicious code between individuals through transfers of computer code between users in an electronic setting.

Consider the problem of prevention of hospital-to-hospital transfer of drug resistant bacteria. Critically ill patients are frequently and routinely transferred between hospitals in order to provide necessary specialized care [15]. While such interhospital transfers are an essential part of routine patient care, they also enable the transfer from hospital to hospital of highly virulent micro-organisms resistant to many or all antibiotics [23]. As an example, recent work [24] implicates inter-hospital patient transfers as an important vehicle for spread of "superbug" MRSA (methicillin-resistant Staphylococcus aureus). There is no existing technology, short of isolating a hospital, that can completely prevent the spread of such micro-organisms. To disrupt transfers by removing a hospital from the system can only be done under truly extraordinary circumstances (such as the outbreak of SARS in Toronto [33]). Instead, there are large numbers of infection control technologies (e.g., bottles of disinfectant) that offer partial prevention and can be applied at individual hospitals (e.g. [38]). Since such infection control technologies are costly, how should policy-makers optimally deploy them in order to minimize the global interhospital spread of

*Dept. of Computer Science, Virginia Tech.
†School of Information, Univ. of Michigan
‡Dept. of Internal Medicine, Univ. of Michigan Medical School
§Dept. of Computer Science, City University of New York
¶Computer Science Dept., Carnegie Mellon University



(a) Connections (b) UNIFORM (∼ current) (c) DEGREE (d) SMART-ALLOC

Figure 1: Our proposed SMART-ALLOC method minimizes number of infections (red circles): (a) The US-MEDICARE network overlayed on a map (b-d) Infected hospitals after a year (365 days) under different immunization algorithms. The same amount of resources ($k = 200$) were distributed by the algorithms. UNIFORM is the largely current practice of distributing evenly across all hospitals, while DEGREE distributes according to the number of connections of a hospital.

highly resistant micro-organisms via patient transfers?

We also consider the spread of malicious content in an electronic setting. In the Second Life virtual world, nearly all content, from the landscape, to clothing, to the avatars' movements, are created and distributed as scripts by the users themselves. This is part of the interactivity that has made the enterprise a success. However these virtual environments create the potential for malicious scripts to be inadvertently picked up and dispersed by unwitting users [1]. Without shutting down a users' account, it is not possible to confer complete immunity upon that node. However, one could allocate resources differentially to a subset of nodes, in the form of educating users and auditing their code inventories.

Motivated by the above applications, and the many other instances where complete immunization is not feasible (e.g. HIV transmission, or H1N1 flu transmission prior to availability of vaccine) we study the problem of effective and efficient fractional immunization on directed weighted graphs. In fractional immunization, one allocates differing amounts of resource to nodes from a fixed total budget. Nodes which receive more infection-prevention resource have a smaller likelihood of becoming infected when exposed than nodes receiving no or little resource. A straightforward approach that tests each possible allocation would very quickly become computationally intractable (e.g., for a network with 2000 nodes, it will take more time than the age of the universe to examine all the possibilities on a 2GHz processor machine). Instead, we give an effective and efficient *linear-time* (in nodes and edges) algorithm SMART-ALLOC in this paper. Our extensive experiments show that we may achieve significant benefits if nodes coordinate their allocation of resources, rather than making allocations independently, as is current practice in many settings. See Figure 1 for an illustration, where our algorithm outperforms other alternatives by up to 6x fewer infections.

The rest of the paper is organized as follows: § 2 reviews related work, § 3 gives the problem formulation and the hardness result, § 4 and § 5 explain our proposed method and § 6 presents extensive experiments on datasets. Finally, we conclude the paper in § 7.

2 Related Work

In short, all the existing immunization strategies mentioned below assume: (1) *full* immunity - once a node is immunized, it is completely removed from the graph; (2) *binary* allocation (i.e., each node would need at most 1 antidote); and (3) *symmetric* immunization - once applied, an antidote affects both incoming and outgoing edges. These assumptions might be too strong for the inter-hospital patient transfers applications. To the best of our knowledge, we are the *first* to address the more realistic and challenging setting, where the effect of an antidote could be partial and asymmetric and the same node can receive multiple antidotes. In addition, w.r.t. the medical world, we are the first to use truly *nation-scale* network data and study alternative approaches to fractional immunization.

We review related work in the context of networks here, which can be categorized into three parts: epidemic thresholds, immunization algorithms and information diffusion.

Epidemic Thresholds/Conditions Much work has been done in finding epidemic thresholds (minimum virulence of a virus which results in an epidemic) for a variety of networks [3, 25, 2, 17, 27, 36, 8, 28, 26]. It should be pointed out that, with the exception of [26] most if not all of the existing work assumes *symmetry* in virus propagation. That is, the probability that A infects B or B infects A is the same, assuming either A or B are infected. The inter-hospital patient transfer graph is *asymmetric*; a hospital that is better equipped to treat a critical care patient is more likely to be on

the receiving end of a transfer. Asymmetries in transfer are also present in e.g. email networks.

Immunization Cohen et al. [6] studied the *acquaintance* immunization policy, and showed that it is much better than random, for both the SIS as well as the SIR model on random power-law graphs. Hayashi et al. [14] modeled e-mail viruses and studied random and targeted immunization. [35] proposed an effective immunization strategy in the SIS model also motivated by preventing the spread of computer viruses, while [9] give a simplified strategy for the IC model. Briesemeister et al. [5] studied immunization policies on power-law graphs. There also has been work on remotely related problems like 'outbreak-detection' [22] and finding most-likely 'culprits' of an infection [20, 29]. In the medical literature, most theoretical work on the spread of highly resistant organisms has focused on spread within a single hospital or population [37].

General information diffusion processes There is a lot of research interest in studying dynamic processes on large graphs, (a) blogs and propagations [13, 19, 16, 30], (b) information cascades [4, 10, 12] and (c) marketing and product penetration [31, 21]. These dynamic processes are all closely related to virus propagation. For example, one may wish to allocate third-party "fact checking" resources to content posted on specific blogs in order to minimize the spread of misinformation in the network. Although no blog could be completely immune to spreading misinformation, such efforts would dampen its spread.

3 Problem Formulation and Hardness result

We first formulate the problem explicitly. Let A be the adjacency matrix of the directed weighted graph (of N nodes and E edges) on which the virus is spreading—entry $A(i, j)$ in the matrix denotes the weight on the edge between nodes (hospitals) i and j (e.g., the weight can be the average number of patient-transfers per day). We assume an infection can in principle reach any node from any other node (i.e. the graph is strongly-connected). The infection spreading model can be described by a flu-like model with no immunity, technically the SI model ('susceptible-infected') of epidemiology [2] (as once infected, it is very hard to completely eradicate a drug-resistant strain). Briefly, at every time-step, any healthy node can get the infection from one of its currently infected neighbors. The probability of becoming infected by any particular neighbor during a period of time is independent and proportional to the weight of the connecting edge. Also, once infected, a node always stays infected. Any node gets partial immunity upon getting an antidote. Any amount x of antidote cuts the transmissibility of the virus by a factor $f(x)$ (called the utility function). For example, under function $f(x) = 0.5^x$, each additional antidote to hospital i decreases the probability of transmission from any neighbor j of i by a fixed percentage (50%). Our results hold for any utility function $f(x)$ with a diminishing marginal returns property typical of infection-control techniques (c.f. [38]). Also note the inherent *asymmetric* nature of the effect of an antidote, it only effects the incoming edges of any node. The infection starts with some initially infected 'seed' nodes. We want to distribute the antidotes so that the expected "footprint" (the expected number of infections at some given time t) is minimized. To summarize, we are given:

- The SI model as the virus-spreading process
- A fixed directed weighted connected graph A (each edge e having weight w_e with $0 < w_e \leq 1$ e.g., the weight can be the average number of patient-transfers per day between hospitals)
- A total of k antidotes having partial effect e.g., bottles of disinfectant
- A weakening function $f(x)$, denoting how beneficial are x units of antidote, typically with diminishing marginal returns property

Using popular epidemiological assumptions, we assume that the virus and the underlying graph do not change.[1] The problem can be stated as:

PROBLEM 1. (MAX-HEALTH) *Distribute the antidotes such that for an infection process spreading over the resulting graph after the antidote-allocation, we minimize the "footprint" (the expected number of infections at time t, for some given t).*

The current practice in allocating varying amounts of antidote across a network is effectively uniform, e.g. hospitals independently tackle infection control [32]. However, this makes no use of the connected network we are given. As mentioned in the introduction, a computationally prohibitively expensive method is to estimate the footprint through computer simulations. How can we get a practical and effective algorithm?

3.1 Our proposed problem—MIN-CONN

Main Idea Our observation is that the footprint depends on the connectivity of the underlying network and as we show next, the best single measure of connectivity is λ_A, the so-called 'largest eigenvalue' of the adjacency matrix of the network. Roughly, it describes the number of paths between pairs of nodes in a graph, discounting for longer paths. Earlier results [36, 8, 28] have also shown that the epidemic threshold (maximum virus

[1] Relaxing these assumptions is a promising research direction.

strength so that there is no epidemic) on unweighted, undirected graphs depends on the largest eigenvalue of the adjacency matrix. Instead of MAX-HEALTH, we then propose to minimize the largest eigenvalue of the weighted adjacency matrix while distributing the antidotes.

Justification Unfortunately, note that unlike other models, our virus spreading model is SI and thus has *no* epidemic threshold - any initial infection will eventually infect *everyone* in the graph. But still, as our next lemma shows, we can upper-bound the expected number of infected nodes in the graph at any time t:

LEMMA 1. *In the SI virus spreading model on a graph:*

$$\sigma(t) \leq (1 + \lambda_A)^t \sigma(0)$$

where $\sigma(t)$ is the expected num. of infected nodes at time $t > 0$ and $\sigma(0)$ is a scalar depending just on the initial conditions (independent of t).

Proof. In the appendix.

Thus, we propose to minimize the upper-bound on the expected number of infected nodes at any time t, by minimizing the largest eigenvalue λ_A.

We call our proposed problem MIN-CONN. Suppose the vector which gives us the distribution for k antidotes is $\vec{m} = \{m_1, m_2, \ldots, m_N\}$ (where m_i is the number of antidotes given to node i), with the constraint that $\sum m_i = k$. Denoting A' as the resulting adjacency matrix after distributing the antidotes, our transformed problem can be stated as:

PROBLEM 2. (MIN-CONN) *Distribute the antidotes such that the largest eigenvalue of the resulting adjacency matrix is minimized i.e.*

$$\text{minimize } \lambda_{A'} \quad \text{s.t.} \quad \sum_i m_i = k, \ \forall_i \ m_i \in \{0, 1, ..\}$$

It is easy to see that if we define a matrix $F = \text{diag}(f(\vec{m}))$ ($f(\vec{m})$ just applies the function f on each element of the vector $\vec{m}$), then $A' = A \times F$.

3.2 MIN-CONN is NP-complete Unfortunately, MIN-CONN is NP-complete. Consider the decision version of MIN-CONN:

PROBLEM 3. (MIN-CONN DECISION VERSION) *Given a directed and weighted graph $G = (V, E)$, $k \geq 0$, $t \geq 0$, and non-increasing $f(x)$ (hence, instance $(G, k, t, f(x))$) is there an assignment $\vec{m}$ with $\sum_i m_i = k, \forall_i \ m_i \in \mathbb{Z}^*$ such that $\lambda_{AF} \leq t$ where A is the adjacency matrix of G and $F = \text{diag}(f(\vec{m}))$?*

We will prove MIN-CONN (Decision version) is NP-complete next.

THEOREM 1. *MIN-CONN (Decision Version) is NP-complete.*

Proof. In the appendix.

4 Proposed Method—Overview

As MIN-CONN is NP-complete, we resort to heuristics. A simple and intuitive heuristic is to disrupt the connectivity of the network by distributing the antidotes according to the number of neighbors ('degree') of a hospital. Thus a hospital involved in a larger number of total patient transfers will receive more resources than small isolated hospitals. This appears to be a reasonable approach until we realize that this does not directly attack the exact connectivity metric: λ_A. For example, this method will allocate most of the resources to the big coastal hospitals, and may miss out on a critical but mid-sized central hospital acting as a 'bridge' between the coasts. Hence, our heuristic should directly try to optimize the drop in λ_A. Next we present two such heuristics in improving speed: (a) `EXHAUSTIVE`, (b) `SMART-ALLOC`.

4.1 Algorithm `EXHAUSTIVE` Algorithm `EXHAUSTIVE` greedily tries to find the best hospital to allocate each additional antidote to. Clearly, the best node is the one in the graph which, when given the extra antidote, decreases λ_A the most at that step. Hence, we need to compute the largest eigenvalue N times for making only a *single* allocation decision (so for k antidotes, it will be done $k \times N$ times). This is very expensive e.g., for a US-wide network of about 2000 hospitals, it took about a *day* to distribute only 1500 antidotes. The total running time would $O(kNE)$ (using the Lanczos algorithm for computing the eigenvalue which is $O(\#edges)$ for sparse graphs). For larger graphs (such as our Second-Life network), this would be too slow to be feasible.

4.2 Algorithm `SMART-ALLOC` We give an overview of our approach here, and the theoretical under-pinnings in the next section.

Best single allocation Following from the discussion above, instead, we can give each additional antidote to the currently most 'important' (central) hospital, with the hope that it is also the hospital reducing λ_A the most. Fortunately, we can show that the measure of centrality which allows us to closely approximate the drop in λ_A is the so-called Eigenvector centrality adapted to directed graphs (a combination of the so-called 'hub'-ness and 'authority'-ness scores [18] of each node). We just give the next antidote to the hospital

which has the highest such centrality score currently. This would be faster than `EXHAUSTIVE`, though with some approximation. Note that we still have to perform the eigenvalue computation (to update the centralities of all the nodes) after each allocation decision. Can we do better?

Batched allocation The answer is yes - in fact, we can make r times fewer updates (for a suitably chosen r) to node centralities by carefully allocating r antidotes in one go, using *only* the old centrality values. Thus we need to update and 'resync' the centralities only every r allocations. We call this algorithm `SMART-ALLOC`: it is much faster (linear on number of nodes and edges) than the other methods with minimal loss of accuracy at the same time (a point we illustrate using experiments as well—see Sections 6.2 and 6.3).

5 Proposed Method—Theorems and proofs

Here we give details of the two main ideas we mentioned above. Jumping ahead, our effective and efficient algorithm `SMART-ALLOC` is given in § 5.2.2.

5.1 Best single allocation—Details Let $\vec{u} = [u_1, u_2, \ldots, u_N]^T$ and $\vec{v} = [v_1, v_2, \ldots, v_N]^T$ be right and left eigenvectors of A corresponding to λ_A. In a nutshell, the best node i^* to give a single antidote is the one with the maximum value of $u_i v_i$ i.e. $i^* = \arg\max_i u_i v_i$. We can prove the following lemma to justify it.

LEMMA 2. *Assuming the current adjacency matrix is A, the change in the in the largest eigenvalue $\Delta\lambda_A$ after distributing one antidote to a node, say i, approximated to the first order is given by:*

$$\Delta\lambda_A = \lambda_A \left(\frac{f(1) u_i v_i}{v^T u} - 1 \right)$$

Proof. In the appendix.

This requires the computation of $\vec{u}$ and $\vec{v}$, which is $O(E)$. We can continue giving the antidotes in this way, but as discussed above, we will need to re-compute $\vec{u}$ and $\vec{v}$ after each allocation decision.

5.2 Batched allocation—Details In sum, `SMART-ALLOC` uses Algorithm 1 to batch-allocate and resync till we have exhausted total budget k (see § 5.2.2). We now show how we can batch-allocate r antidotes in one-go. Suppose the distribution of allocations as before is given by the vector $\vec{m}$. In this case, we can prove the following lemma, similar to Lemma 2.

LEMMA 3. *The change in the largest eigenvalue $\Delta\lambda_A$ after distributing r antidotes according to $\vec{m}$ (so $\sum_i m_i = r$) approximated to the first order is given by:*

$$\Delta\lambda_A = \lambda_A \left(\frac{v^T F u}{v^T u} - 1 \right)$$

where v and u are the left and right eigenvectors of A corresponding to λ_A and $F = \mathrm{diag}(f(\vec{m}))$.

Proof. In the appendix.

Subsequently, for the best allocation of r antidotes, it is easy to see that we have the following optimization problem now, analogous to MIN-CONN:

PROBLEM 4. (MAX-DROP) *Distribute antidotes such that:*

$$\text{minimize} \sum_{i=1}^{N} f(m_i) \cdot u_i \cdot v_i \quad \text{s.t.} \quad \sum_i m_i = r$$

(of course, $\forall_i\ m_i \in \{0, 1, ..\}$). Clearly, it is an integer optimization problem, which in general is NP-complete.

5.2.1 *GreedyDrop*: An optimal greedy algorithm Consider the following greedy algorithm: intuitively, at each iteration, we pick the index (node) j which maximizes the drop in the value of the objective *at that* step (see Algorithm 1).

Algorithm 1 *GreedyDrop*

Input: Directed Weighted Adjacency matrix A, batch-size r, monotone non-increasing convex function $f(x)$

1: u = first right eigenvector of A
2: v = first left eigenvector of A
3: $\vec{m} = \vec{0}$
4: **for** $i = 1$ to r **do**
5: $\quad j = \max_h\ [f(m_h) - f(m_h + 1)] u_h v_h$
6: $\quad m_j = m_j + 1$
7: **end for**
8: **return** $\vec{m}$

Surprisingly, we can in fact prove that this achieves the optimal solution for MAX-DROP, when $f(x)$ is *monotone non-increasing convex*. Clearly, the running time of the algorithm is $O(E + kN)$. We prove the optimality of *GreedyDrop* in Theorem 2.

THEOREM 2. *GreedyDrop returns the optimal integral $\vec{m}$ for MAX-DROP when $f(x)$ is monotone non-increasing and convex.*

Proof. In the appendix.

5.2.2 SMART-ALLOC Finally, we are ready to describe our algorithm SMART-ALLOC: use *GreedyDrop* (Algorithm 1) to batch-allocate a small number (r) of resources and then 're-sync' (re-compute) the first left and right eigenvectors and continue similarly till our budget k is exhausted.

One may ask why can not we directly allocate all k antidotes in one-go using *GreedyDrop*? This is because, unfortunately, the accuracy of the first-order approximation in Lemma 3 is only good when the number of antidotes k is small w.r.t. the graph i.e when $k \ll N$. But that is not the case in general - for e.g. in our motivating problem one may want to distribute 200 infection control resources among 2000 nationwide hospitals. In fact, k can be arbitrarily high, since the units of resource in this problem are set with arbitrary granularity. It is easy to see the next lemma:

LEMMA 4. (RUNNING TIME OF SMART-ALLOC) *The running time of the algorithm SMART-ALLOC is $O(kE/r + kN)$ i.e. linear in the size of the graph.*

Clearly, we want to use as large r as possible. Our proposed rule-of-thumb is to choose r proportional to the spectral-gap ($|\lambda_A| - |\lambda_{A,2}|$) of the graph. Larger the spectral-gap, lesser is the sensitivity of the spectrum of A [11], lesser is the need to re-sync often and hence larger is the r we can use e.g. in our experiments on hospital graphs, which had a small spectral-gap, we found that $r = 6$ performed very well.

6 Experiments

We designed experiments to answer the following questions about our algorithm SMART-ALLOC: (i) Effectiveness for reducing the rate of infection, (ii) Effectiveness for MIN-CONN and (ii) Scalability. In short, SMART-ALLOC proves to be a fast and effective algorithm for both reducing the rate of infection and solving MIN-CONN and is very close to EXHAUSTIVE, at a fraction of the running cost, while others are much worse.

6.1 Setup For answering the above questions we ran extensive simulation experiments and compared against many other resource allocation methods (see Table 1) on multiple real-world datasets (see Table 2). We ran parallel experiments on a Condor [34] cluster of 58 cores each being a generic Fedora 7 machine. All the algorithms and the SI infection process were coded in C++. We use $f(x) = 0.50^x$ and $r = 6$ for all our experiments. The choice of the function $f(x)$ captures the diminishing marginal utility of infection control based on a wide-range of studies in the medical literature of existing infection control techniques (c.f. [38]).

Table 1: Various Algorithms used for comparison

Method Name	Method Description	Speed $O(\cdot)$
UNIFORM	Distribute uniformly among the nodes, breaking ties randomly.	kN
DEGREE	Distribute randomly proportional to the 'degree'† of the nodes.	$E + kN$
EXHAUSTIVE	Allocate each additional antidote to that node which decreases the largest eigenvalue λ_A the most in that step.	kEN
SMART-ALLOC	Allocate r antidotes in one go based on node centralities and only then recompute.	$kE/r + kN$

† As the graphs are directed, we use degree centrality [7] - geometric mean of in-degree (the number of transfers the hospital receives) and out-degree (the number of transfers the hospital sends out).

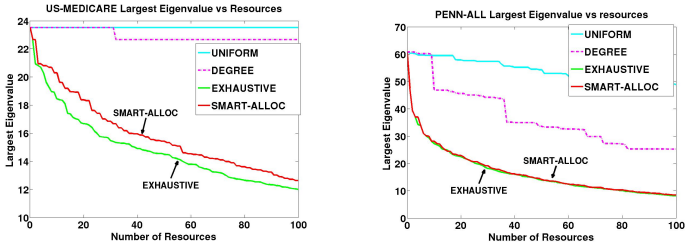
6.2 Effectiveness for MIN-CONN problem MIN-CONN aims to decrease the largest eigenvalue the most - how do the algorithms perform in that measure? In short, SMART-ALLOC comes very close to EXHAUSTIVE while others are much worse. Figure 2 shows the largest eigenvalue of the resulting graph after giving k antidotes according to various algorithms vs k on the US-MEDICARE and PENN-ALL networks. UNIFORM and DEGREE perform poorly, although DEGREE is better (sometimes marginally) than UNIFORM. SMART-ALLOC and EXHAUSTIVE are much better at achieving the lowest eigenvalue for all k. EXHAUSTIVE is expected to be near-optimal as it does an exhaustive search via repeated eigenvalue computation for the node which decreases the eigenvalue the most. On the other hand, SMART-ALLOC performs well due to our careful approximation and algorithm-design.

6.3 Effectiveness for MAX-HEALTH problem We ultimately want to test how the algorithms perform for MAX-HEALTH. In short, again, as also demonstrated by Figure 1 in the introduction, SMART-ALLOC proves to be an effective algorithm and is very close to EXHAUSTIVE while others are much worse. See Figures 3 and 4 - they show the expected number of infected nodes (hospitals) vs. time tick after running the infection process on the partially immunized US-MEDICARE and

Table 2: Various real-world datasets used in our work

Dataset Name	Nodes (N)	Edges (E)	Description
US-MEDICARE	2138	10241‡	All critical patient transfers among US hospitals based on all Medicare Provider Analysis and Review (MedPAR) final action claims between Sept. 1, 2004 - Sept. 1, 2005 [15]. See also Fig. 1(a).
PENN-ALL	137	1121‡	Critical patient transfers within Pennsylvania hospitals based on all discharges (not just Medicare) between July 1, 2004 - June 30, 2006 [15].
GESTURE	166,774	1.5 million	Second-Life transfer-network of 'gestures' among users. Gestures can include anything from animation, chat to playing sounds.

‡Weight for each edge $u \to v$ was the average number of transfers from hospital u to v per day.



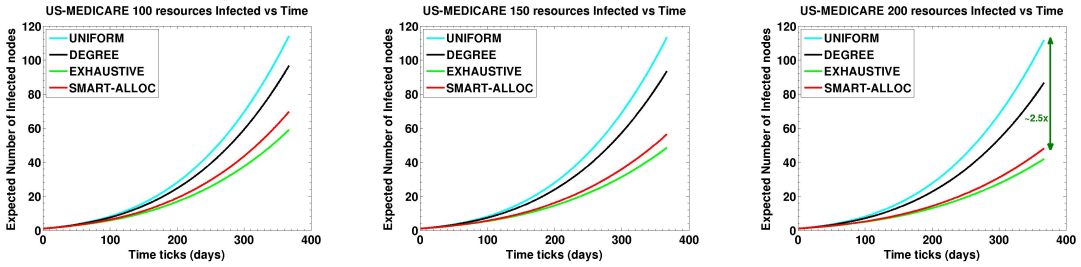
(a) US-MEDICARE (b) PENN-ALL

Figure 2: Largest Eigenvalue after allocation vs Budget k of resources used for different algorithms. (a) US-MEDICARE Network (b) PENN-ALL Network. Lower is better and SMART-ALLOC is near-optimal in both cases. (plots use color)

PENN-ALL networks for different budget k of antidotes. The different curves represent the different algorithms used for allocation. As the edge-weights represent the average number of transfers per day, the curves represent the average footprint for each day after the infection starts. Each curve is an average of 21380 and 1370 simulation runs for US-MEDICARE and PENN-ALL respectively - in this way we ensured that we seeded the infection from each hospital for 10 different runs. We ran the simulations till 365 time-ticks (= 1 year) and took the average over all runs for each time-tick. Finally, the range of values of k for US-MEDICARE and PENN-ALL were chosen according to the network sizes and the function $f(x) = 0.50^x$. Our algorithm SMART-ALLOC clearly is very close to EXHAUSTIVE and has the lowest footprints everyday compared to the rest. For e.g. in Figure 3(c), after an year with $k = 200$ antidotes, EXHAUSTIVE and SMART-ALLOC have an expected total of 42 and 46 hospitals infected, while the other methods end with about 2.5 times worse at around 110. It is even more pronounced in PENN-ALL (Figure 4(c)): after an year with $k = 120$, EXHAUSTIVE and SMART-ALLOC have an expected footprint of ~ 8, while the next closest method is about 3 times worse at around 23. This shows the dramatic impact an effective allocation algorithm can have on the number of infected nodes. Moreover note that all algorithms essentially mimic their performance w.r.t. MIN-CONN (Figure 2) i.e. larger the corresponding drop in the first eigenvalue λ_A, lower is the number of expected infections, validating our reduction of MAX-HEALTH to MIN-CONN.

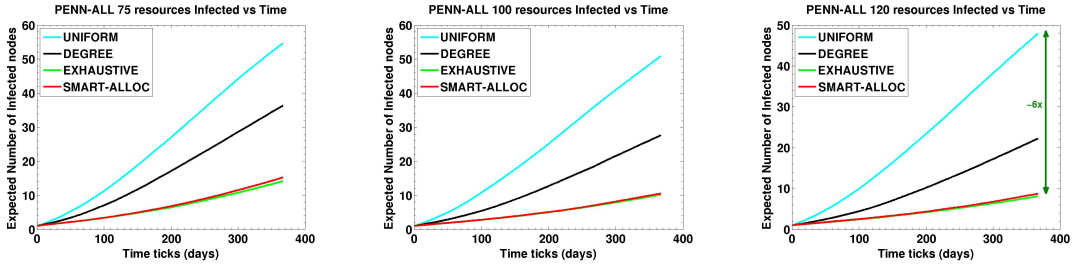
The current practice is for each hospital to independently manage infection control [32], which may be no better from the network perspective than using UNIFORM. But note that compared to UNIFORM, SMART-ALLOC can be up to 6 times better (see Figure 4(c)). Interestingly, for the US-MEDICARE network, we found that to achieve the same level of infection control as SMART-ALLOC and $k = 120$, we need a budget of about $k = 800$ resources if distributed according to UNIFORM.

6.4 Scalability As discussed before, SMART-ALLOC is much faster than its chief competitor EXHAUSTIVE (see Table 1). For example, it took more than *10 hours* to distribute 200 resources using EXHAUSTIVE on the US-MEDICARE network while it took just $\sim$ *14 seconds* to run SMART-ALLOC for the same budget - a 2500x speed-up. As a further comparison, the naïve simulation-based algorithm ran for a week and still had not finished for the same budget - a more than 30,000x speed-up. Additionally, on the GESTURE network, we had to stop



(a) $k = 100$ (b) $k = 150$ (c) $k = 200$

Figure 3: US-MEDICARE network for different algorithms and budget k of resources: Expected Number of Infections vs Time ticks ($\approx$ days). Again `EXHAUSTIVE` and `SMART-ALLOC` perform the best and are close to each other, as expected. Each curve average of 21380 runs and lower is better (plot uses color)



(a) $k = 75$ (b) $k = 100$ (c) $k = 120$

Figure 4: PENN-ALL network for different algorithms and budget k of resources: Expected Number of Infections vs Time ticks ($\approx$ days). Again `EXHAUSTIVE` and `SMART-ALLOC` perform the best (they are almost on top of each other), as expected. Each curve average of 1370 runs and lower is better (plot uses color)

`EXHAUSTIVE` after it took *3 days* to allocate a *single* resource; `SMART-ALLOC` took $\sim$ 150 mins to allocate *2000* resources.

6.5 Generality To demonstrate the utility of `SMART-ALLOC` in many other scenarios other than epidemiology, we also compare performance on the GESTURE network of asset transfers between virtual world users. More details and a plot can be found in the appendix; in short, `SMART-ALLOC` has the fewest users infected, while others have up to $\sim$ 2.5 times more users infected. Also, `EXHAUSTIVE` didn't even complete after *3 days* whereas `SMART-ALLOC` allocated 2000 resources in only $\sim$ 150 mins, demonstrating its superiority.

7 Conclusion

This paper addresses the problem of allocation of infection-control resources with fractional and asymmetric impact among nodes in a network. It is a more general problem than that of selecting a subset of nodes to be immunized completely via a vaccine. The potential applications are broad—from curbing spread of infection between hospitals from patient transfers, to preventing spread of malicious code in virtual world settings.

We formulated the problem, proved it is NP-complete, and gave a highly efficient and effective algorithm `SMART-ALLOC`, which we also demonstrated through extensive experiments on multiple real-world datasets, including nation-wide patients-transfer networks and electronic virtual-world social transfer-networks. `SMART-ALLOC` runs in seconds (as opposed to weeks), on commodity hardware; more importantly, applied on real hospital-transfer networks (2005 U.S. Medicare data, 2004-2006 PA all-payer data) it results to up to *6x* fewer infections, compared to current practice and other heuristics.

The current practice in control of highly resistant organisms via patient transfers has been largely focused within individual hospitals. Hence, the public health policy is missing an opportunity to significantly reduce infection rates with an infection prevention strategy that accounts for the potential transfer of bacteria along the network of inter-hospital patient transfers.

Acknowledgments: This material is based upon work supported by the U.S. Army Research Laboratory under Cooperative Agreement No. W911NF-09-2-0053, the DARPA under Agreement No. W911NF-12-C-0028, the NIH under Grant No. K08, HL091249 and the NSF under Grants No. IIS-1017415 and IIS-0746646.

References

[1] Url:http://www.reuters.com/article/2007/08/20/us-disease-game-idusn2031054020070820?sp=true.

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

[3] N. Bailey. *The Mathematical Theory of Infectious Diseases and its Applications*. Griffin, London, 1975.

[4] S. Bikhchandani, D. Hirshleifer, and I. Welch. A theory of fads, fashion, custom, and cultural change in informational cascades. *Journal of Political Economy*, 100(5):992–1026, October 1992.

[5] L. Briesemeister, P. Lincoln, and P. Porras. Epidemic profiles and defense of scale-free networks. *WORM 2003*, Oct. 27 2003.

[6] R. Cohen, S. Havlin, and D. ben Avraham. Efficient immunization strategies for computer networks and populations. *Physical Review Letters*, 91(24), Dec. 2003.

[7] L. C. Freeman. Centrality in social networks: Conceptual clarification. *Social Networks*, 1:215–239, 1979.

[8] A. Ganesh, L. Massoulié, and D. Towsley. The effect of network topology on the spread of epidemics. In *INFOCOM*, 2005.

[9] G. Giakkoupis, A. Gionis, E. Terzi, and P. Tsaparas. Models and algorithms for network immunization. *Technical report*, 2005.

[10] 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.

[11] G. H. Golub and C. F. Van-Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, 2nd edition, 1989.

[12] M. Granovetter. Threshold models of collective behavior. *Am. Journal of Sociology*, 83(6):1420–1443, 1978.

[13] D. Gruhl, R. Guha, D. Liben-Nowell, and A. Tomkins. Information diffusion through blogspace. In *WWW '04*, 2004.

[14] Y. Hayashi, M. Minoura, and J. Matsukubo. Recoverable prevalence in growing scale-free networks and the effective immunization. *arXiv:cond-mat/0305549 v2*, Aug. 6 2003.

[15] T. J. Iwashyna, J. D. Christie, J. Moody, J. M. Kahn, and D. A. Asch. The structure of critical care transfer networks. *Medical Care*, 47(7):787–793, 2009.

[16] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *KDD*, 2003.

[17] J. O. Kephart and S. R. White. Measuring and modeling computer virus prevalence. *IEEE Symp. on Research in Security and Privacy*, 1993.

[18] J. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46, 1999.

[19] R. Kumar, J. Novak, P. Raghavan, and A. Tomkins. On the bursty evolution of blogspace. In *WWW*, 2003.

[20] T. Lappas, E. Terzi, D. Gunopoulos, and H. Mannila. Finding effectors in social networks. *SIGKDD*, 2010.

[21] J. Leskovec, L. A. Adamic, and B. A. Huberman. The dynamics of viral marketing. In *EC*, pages 228–237, New York, NY, USA, 2006. ACM Press.

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

[23] D. M. Livermore. Bacterial resistance: Origins, epidemiology, and impact. *Clinical Infectious Diseases*, 36(S1), 2003.

[24] P. R. McAdam and et. al. Molecular tracing of the emergence, adaptation, and transmission of hospital-associated methicillin-resistant staphylococcus aureus. *Proc. Natl. Acad. Sc. USA (Early Edition)*, 2012.

[25] A. G. McKendrick. Applications of mathematics to medical problems. In *Proceedings of Edin. Math. Society*, volume 14, pages 98–130, 1926.

[26] L. A. Meyers, M. Newman, and B. Pourbohloul. Predicting epidemics on directed contact networks. *Journal of Theoretical Biology*, 240(3):400 – 418, 2006.

[27] R. Pastor-Satorras and A. Vespignani. Epidemic dynamics in finite size scale-free networks. *Physical Review E*, 65:035108, 2002.

[28] B. A. Prakash, D. Chakrabarti, M. Faloutsos, N. Valler, and C. Faloutsos. Threshold conditions for arbitrary cascade models on arbitrary networks. In *IEEE ICDM*, 2011.

[29] B. A. Prakash, J. Vreeken, and C. Faloutsos. Spotting culprits in epidemics: Who and how many? In *IEEE ICDM*, 2012.

[30] M. Richardson and P. Domingos. Mining knowledge-sharing sites for viral marketing. *SIGKDD*, 2002.

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

[32] D. Smith, S. Levin, and R. Laxminarayan. Strategic interactions in multi-institutional epidemics of antibiotic resistance. *Proc. Natl. Acad. Sc. USA*, 102(8):3153–8, 2005.

[33] T. Svoboda, B. Henry, L. Shulman, E. Kennedy, E. Rea, W. Ng, and et al. Public health measures to control the spread of the severe acute respiratory syndrome during the outbreak in toronto. *New England Journal of Medicine*, 350:2352–61, 2004.

[34] D. Thain, T. Tannenbaum, and M. Livny. Distributed computing in practice: the condor experience. *Concurrency - Practice and Experience*, 17(2-4):323–356, 2005.

[35] H. Tong, B. A. Prakash, C. E. Tsourakakis, T. Eliassi-Rad, C. Faloutsos, and D. H. Chau. On the vulnerability of large graphs. In *IEEE ICDM*, 2010.

[36] Y. Wang, D. Chakrabarti, C. Wang, and C. Faloutsos. Epidemic spreading in real networks: An eigenvalue viewpoint. In *SRDS*, 2003.

[37] G. Webb, E. D'Agata, P. Magai, and S. Ruan. A model of antibiotic-resistant bacterial epidemics in hospitals. *Proc. Natl. Acad. Sc. USA*, 102(37):13343–8, 2005.

[38] W. Zingg, A. Imhof, M. Maggiorini, R. Stocker, E. Keller, and C. Ruef. Impact of a prevention strategy targeting hand hygiene and catheter care on the incidence of catheter-related bloodstream infections. *Crit Care Med.*, 37(7):2167–2173, 2009.

Appendix: Fractional Immunization in Networks

B. Aditya Prakash* Lada Adamic† Theodore Iwashyna‡ Hanghang Tong§
Christos Faloutsos¶

1 Proofs from Section 3

LEMMA 1. *In the SI virus spreading model on a graph:*

$$\sigma(t) \leq (1+\lambda_A)^t \sigma(0)$$

where $\sigma(t)$ is the expected num. of infected nodes at time $t > 0$ and $\sigma(0)$ is a scalar depending just on the initial conditions (independent of t).

Proof. Suppose the discrete-time SI process is running on graph A and $p_i(t)$ denotes the probability that node i is infected at time t after the process started. Then,

$$p_i(t+1) = p_i(t) + (1-p_i(t)) \cdot \Gamma_i \quad (1)$$

where Γ_i is the probability that node i receives some infection from any of its infected neighbors during the time t to $t+1$. Let R be an indicator random variable for the event that node i gets the infection during t to $t+1$. Clearly,

$$R = \mathbb{1}_{\bigcup_{j \in \text{neighbor}(i)} T_j}$$

where T_j is the event that node j transferred an infection between time t and $t+1$; $\mathbb{1}_j(t)$ is the corresponding indicator random variable. Using the well-known relation that expectation of an indicator random variable is just the p.d.f. of the random variable:

$$\begin{aligned} \Gamma_i = \mathbb{E}[R] &= \mathbb{E}[\mathbb{1}_{\bigcup_{j \in \text{neighbor}(i)} T_j}] \\ &\leq \sum_{j=1}^{N} \mathbb{E}[\mathbb{1}_j(t)] = \sum_{j=1}^{N} A(j,i) p_j(t) \end{aligned}$$

where the second step follows because for any two events A and B, $\mathbb{1}_{A \cup B} = \mathbb{1}_A + \mathbb{1}_B - \mathbb{1}_A \mathbb{1}_B \Rightarrow \mathbb{E}[\mathbb{1}_{A\cup B}] \leq \mathbb{E}[\mathbb{1}_A] + \mathbb{E}[\mathbb{1}_B]$. Thus using Equation 1 and above:

$$p_i(t+1) \leq p_i(t) + (1-p_i(t)) \sum_{j=1}^{N} A(j,i) p_j(t)$$

Letting $\vec{P}(t) = [p_1(t), p_2(t), \ldots, p_N(t)]^T$, we can write the entire system as:

$$\begin{aligned} \vec{P}(t+1) &\leq \vec{P}(t) + [I - \text{diag}(\vec{P}(t))] \times A^T \times \vec{P}(t) \\ &= \vec{P}(t) + A^T \vec{P}(t) - \text{diag}(\vec{P}(t)) A^T \vec{P}(t) \\ &\leq (I + A^T) \vec{P}(t) \\ &\leq (I + A^T)^t \vec{P}(0) \end{aligned}$$

Consider the all ones vector $\vec{e}$. Then for any $t > 0$, $\vec{e}^T \vec{P}(t) = \sigma(t)$, the expected number of infected nodes at time t. Hence,

$$\begin{aligned} \sigma(t+1) &\leq \vec{e}^T (I + A^T)^t \vec{P}(0) \\ &= \vec{e}^T (\sum_{j=1}^{N} (1+\lambda_{A,i})^t \vec{v}_i \vec{u}_i^T) \vec{P}(0) \\ &\leq (1+\lambda_{A,1})^t \vec{e}^T (\sum_{j=1}^{n} \vec{v}_i \vec{u}_i^T) \vec{P}(0) \end{aligned}$$

where we used the spectral decomposition of matrix $I + A^T$ in the second step. Denoting $\lambda_{A,1}$ as λ_A, we have that

$$\sigma(t+1) \leq (1+\lambda_A)^t \sigma(0)$$

where $\sigma(0) = \vec{e}^T (\sum_{j=1}^{n} \vec{v}_i \vec{u}_i^T) \vec{P}(0)$ (a scalar depending just on the initial conditions independent of t). □

THEOREM 1. *MIN-CONN (Decision Version) is NP-complete.*

Proof. Clearly, MIN-CONN (Decision Version) is in NP: given an integral assignment $\vec{m}$ as witness, we can check in poly-time if the largest eigenvalue is less than the threshold. Hence we just need to prove that it is poly-time reducible from an NP-complete problem.

We reduce from INDEPENDENT-SET, a well-known NP-complete problem [1].

PROBLEM. (INDEPENDENT-SET) *Given a undirected, unweighted graph $G = (V, E)$ and a number $k > 0$ (i.e. instance (G, k)), is there a set of k vertices, no two of which are adjacent?*

*Dept. of Computer Science, Virginia Tech.
†School of Information, Univ. of Michigan
‡Dept. of Internal Medicine, Univ. of Michigan Medical School
§Dept. of Computer Science, City University of New York
¶Computer Science Dept., Carnegie Mellon University

Say the size of G is n. Given an instance of INDEPENDENT-SET (G, k) we create an instance $I \equiv (G, n-k, 0, f(x))$ of MIN-CONN where $f(x)$ is defined as

$$f(x) = \begin{cases} 1, & \text{if } x = 0 \\ 0, & \text{if } x > 0 \end{cases}$$

Note that such a $f(x)$ forces any algorithm for MIN-CONN to essentially choose k vertices whose all incoming edges will be deleted. Clearly this construction takes polynomial time. We now need to prove two things:
1. If there is an independent set S in G, the instance I has a YES answer.
This is true, because we can set $m_i = 1$ for all $n-k$ nodes i *not* in S (i.e. $V \setminus S$). Consider the resulting graph G'. There will not be any edges from vertices in S to any other vertex. Also, there will not be any edges from vertices in set $V \setminus S$ to each other. These follow because of the antidote distribution and the fact that S was an independent set for G. Hence, the adjacency matrix AF of G' will look like:

$$AF = \begin{bmatrix} 0_{n-k,n-k} & C \\ 0_{k,n-k} & 0_{k,k} \end{bmatrix}$$

where C is a size $(n-k) \times k$ matrix representing the edges from $V \setminus S$ to S. It is easy to check that the largest eigenvalue of AF is 0 (since the lower-triangle part as well as the diagonal elements in AF are all zeros, all the eigenvalues of AF are zeros). Hence I has a YES answer.
2. If G does not have an independent set of size k, then instance I has a NO answer.
Suppose the algorithm for MIN-CONN selects $n-k$ vertices whose all incoming edges will be deleted. Call the un-selected vertices set S ($|S| = k$) and the resulting graph G' (adjacency matrix AF). Consider G_S and G'_S, the subgraph induced by the vertices of S in G and G' respectively. Clearly $G_S \equiv G'_S$, as the algorithm didn't select any vertex in S. Also, as G does not have an independent set of size k, G_S is not a null graph (with no edges) and thus has some connected sub-graph H. Applying the Perron-Frobenius theorem [2], the largest eigenvalue of the adjacency matrix for H is positive. Denoting the adjacency matrix of G'_S (or G_S) as D, the matrix AF will look like:

$$AF = \begin{bmatrix} 0_{n-k,n-k} & C \\ 0_{k,n-k} & D \end{bmatrix}$$

where like before C is a size $(n-k) \times k$ matrix representing the edges from $V \setminus S$ to S. We know that the largest eigenvalue of AF is at least the largest eigenvalue of D and the largest eigenvalue of D is at least the largest eigenvalue of the adjacency matrix of H (eigenvalue interlacing). Hence, D has at least one non-zero eigenvalue. Thus for any S, the largest eigenvalue of AF is non-zero and hence instance I has a NO answer.

Hence, MIN-CONN (Decision version) is NP-complete. □

2 Proofs from Section 4

LEMMA 2. *Assuming the current adjacency matrix is A, the change in the in the largest eigenvalue $\Delta\lambda_A$ after distributing one antidote to a node, say i, approximated to the first order is given by:*

$$\Delta\lambda_A = \lambda_A \left(\frac{f(1) u_i v_i}{v^T u} - 1 \right)$$

Proof. We know that $Au = \lambda_A u$ and $v^T A = \lambda_A v^T$ (right and left eigenvectors). As A is strongly-connected, according to the Perron-Frobenius theorem [2], λ_A is real and non-negative and the components of the corresponding eigenvectors v and u all are positive. After a small modification due to the medicine:

$$(A + \Delta A)(u + \Delta u) = (\lambda_A + \Delta\lambda_A)(u + \Delta u)$$

Premultiplying by v^T and neglecting second order terms:

$$\Delta\lambda_A \approx \frac{v^T \Delta A u}{v^T u} \tag{2}$$

Clearly, after distributing one antidote to node i, ΔA is:

$$\Delta A = AF_i - A \tag{3}$$

where $F_i = diag([f(0), \ldots, f(1), \ldots, f(0)])$ (i.e. the i-th position on the diagonal is $f(1)$). Using it in Equation 2:

$$\begin{aligned} \Delta\lambda_A &\approx \frac{v^T A F_i u}{v^T u} - \frac{v^T A u}{v^T u} \\ &= \frac{\lambda_A v^T F_u}{v^T u} - \lambda_A \\ &= \lambda_A \left(\frac{f(1) u_i v_i}{v^T u} - 1 \right) \end{aligned} \tag{4}$$

Proved. □

LEMMA 3. *The change in the largest eigenvalue $\Delta\lambda_A$ after distributing r antidotes according to $\vec{m}$ (so $\sum_i m_i = r$) approximated to the first order is given by:*

$$\Delta\lambda_A = \lambda_A \left(\frac{v^T F u}{v^T u} - 1 \right)$$

where v and u are the left and right eigenvectors of A corresponding to λ_A and $F = \mathrm{diag}(f(\vec{m}))$.

Proof. (Details Omitted for brevity) The main change from Lemma 2 is that $\Delta A = AF - A$ now. □

THEOREM 2. *GreedyDrop returns the optimal integral $\vec{m}$ for MAX-DROP when $f(x)$ is monotone non-increasing and convex.*

Before proving this theorem, we need the following lemma.

LEMMA 4. *Given a convex non-increasing function $f(x)$, define function $g(x) = f(x) - f(x+1)$. Then $g(x)$ is non-increasing.*

Proof. As $f(x)$ is monotone non-increasing and convex, from the property of convex functions:

$$f(x) - f(y) \geq f'(y)[x - y] \quad \forall x, y \tag{5}$$

Using Equation 5 with $x = x, y = x + 1$ and $x = x + 1, y = x$, we get:

$$-f'(x+1) \leq g(x) \leq -f'(x)$$

Similarly,

$$-f'(x+2) \leq g(x+1) \leq -f'(x+1)$$

Clearly, from the preceding inequalities, we have that $\forall x \;\; g(x+1) \leq g(x)$ i.e. $g(x)$ is a non-increasing function. □

We are now ready to prove Theorem 2.

Proof. Say *GreedyDrop* returns m^G as the answer, but m^* is the true optimal. Then there was some first step (say t) where we incremented some m_j from s_j to $s_j + 1$ in m^G but m^* has $m_j = s_j$. Because we have a fixed batch-budget r, m^* also has some m_k as $s_k + 1$ while m^G has m_k which is *at most* s_k.

Consider another assignment m' which is identical to m^* except $m_k = s_k$ and $m_j = s_j + 1$. Note that we are still satisfying our constraint and hence it is a valid assignment. The score of this assignment is:

$$\begin{aligned} \text{Score}(m') &= \sum_{i=1}^{N} f(m_i) \cdot u_i \cdot v_i \\ &= \text{Score}(m^*) + [f(s_k) - f(s_k + 1)]u_k v_k \\ &\quad -[f(s_j) - f(s_j + 1)]u_j v_j \end{aligned} \tag{6}$$

where the last step is due to the construction of m'.

Recall that while computing m^G, *GreedyDrop* had chosen j at step t i.e.,

$$j = \max_h \; [f(m_h) - f(m_h + 1)]u_h v_h$$

at step t. At that instant, suppose $m_k = s'_k$. Hence from the above equation we can conclude that:

$$[f(s'_k) - f(s'_k + 1)]u_k v_k \leq [f(s_j) - f(s_j + 1)]u_j v_j \tag{7}$$

In addition, we know that $s'_k \leq s_k$. But from Lemma 4, $g(s'_k) \geq g(s_k)$ i.e.

$$f(s'_k) - f(s'_k + 1) \geq f(s_k) - f(s_k + 1) \tag{8}$$

So, from Equations 7 and 8:

$$[f(s_k) - f(s_k + 1)]u_k v_k \leq [f(s_j) - f(s_j + 1)]u_j v_j$$

Coupled with Equation 6, the above inequality implies that $\text{Score}(m') \leq \text{Score}(m^*)$. If $\text{Score}(m') < \text{Score}(m^*)$, then m^* is not optimal as we started with the assumption that m^* is optimal and hence has the lowest score. If $\text{Score}(m') = \text{Score}(m^*)$, then we can conclude that *GreedyDrop* did not make an error at step t and made it at some other point. Continuing similarly, finally, either m^* is not optimal or *GreedyDrop* is correct. Hence, a contradiction, m^G is optimal and *GreedyDrop* gives the optimal integral answer. □

3 Additional Experiments: Generality

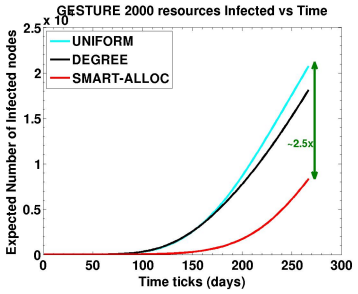


Figure 1: Expected number of infections vs time-ticks for different algorithms, budget $k = 2000$ on the GESTURE network. `SMART-ALLOC` is the best. Each curve average of 1000 runs. (plot uses color)

As mentioned in the introduction of the paper, although our problem was originally motivated on hospital-transfer networks, the problem of fractional immunization arises in many other scenarios, and is arguably more realistic than complete immunization. To demonstrate the utility of `SMART-ALLOC` in domains other than epidemiology, we also compare performance on the GESTURE network of asset transfers between virtual world users. Figure 1 (in the appendix) shows the the expected number of infected users vs. time, if a malicious asset were to be created and transferred between users. We budget $k = 2000$ antidotes, and

select the infection source randomly. We don't show `EXHAUSTIVE` because, as mentioned before, it didn't complete even after *3 days* whereas `SMART-ALLOC` allocated all 2000 resources in $\sim$ 150 mins. As expected, `SMART-ALLOC` has the fewest users infected, while others have up to $\sim$ 2.5 times more users infected, demonstrating the efficacy of our algorithm in a completely different domain.

References

[1] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman, 1983.

[2] C. R. McCuler. The many proofs and applications of perron's theorem. *SIAM Review*, 42, 2000.