Limiting the Spread of Epidemics within Time Constraint on Online Social Networks

Canh V. Pham
University of Engineering and Technology, Vietnam National University
Hanoi, Vietnam
Faculty of Information Technology and Security, People's Security Academy
Hanoi, Vietnam
maicanhki@gmail.com

Hoang M. Dinh, Hoa D. Nguyen, Huyen T. Dang
Faculty of Information Technology and Security, People's Security Academy
Hanoi, Vietnam
dinhmanhhoang150197@gmail.com, hoand.hvan@gmail.com, danghuyen1997@gmail.com

Huan X. Hoang
University of Engineering and Technology, Vietnam National University, Hanoi, Vietnam
Hanoi, Vietnam
huanhx@vnu.edu.vn

ABSTRACT

In this paper, we investigate the problem of limiting the spread of epidemics on online social networks (OSNs) with the aim to seek a set nodes of size at most k to remove from the networks such that the number of saved nodes is maximal for cases where we already know the set of infected nodes on the networks. The problem is proved to be NP-hard and it is NP-hard to approximate the problem with ratio $n^{1-\epsilon}$, for $0 < \epsilon < 1$. Besides, we also suggest two algorithms to solve the problem. Experimental results show that our propsed outperform baseline algorithms.

CCS CONCEPTS

• **Networks** → Network algorithms;

KEYWORDS

Online Social Networks, information difussion, epidemics

ACM Reference Format:
Canh V. Pham, Hoang M. Dinh, Hoa D. Nguyen, Huyen T. Dang, and Huan X. Hoang. 2017. Limiting the Spread of Epidemics within Time Constraint on Online Social Networks. In *SoICT '17: Eighth International Symposium on Information and Communication Technology, December 7–8, 2017, Nha Trang City, Viet Nam.* ACM, New York, NY, USA, Article 4, 8 pages. https://doi.org/10.1145/3155133.3155157

1 INTRODUCTION

With billions of users, Online Social Networks (OSNs), such as Facebook, Twitter and Google+, OSNs have provided an effective platform for connection and communication among people. Unfortunately, this feature can be used to spread the epidemics quickly on OSNs. Epidemics can be bad factors on OSNs, such as: virus, rumors, misinformation, etc. The field of preventing the spreading of epidemics has received a lot of research interests in recent years. One of the important works to against the spread of epidemics is detecting the sources. Information about the content of the posts, comments, shares may be collected to detect epidemic sources. Qazvinian et al. [23] proposed an effective solution for detecting source based on content, network and microblog-specific memes. Rumors are identified through the use: temporal, structural, and linguistic [13]. Nguyen et al. found the solution for the problem of identifying the set of at most k-suspected users from the set of victims who are influenced by the misinformation [21]. Recently, for limitting of epidemics propagation, a general strategy is to block accounts and links that play an important role in process of influence of epidemics [12, 25]. However, in reality, we can not exclude too many nodes and links that play an important role in misinformation/rumors difussion more effectively. In order to minimize the influence of known misinformation sources, there were some heuristic methods to remove edges was disigned [10, 11]. Khalil et al. [9] proposed the problem of removing the set of edges to minimize the influence of the source of information in LT model. Recently, Zhang et al. designed strategies to decontaminate the effects of harmful sources by vaccinating nearby nodes that is equivalent to removing the nodes from the graph [28, 29]. However, their algorithms are not theoretically guaranteed. Zhang et al. [26] proposed placement monitors strategy to prevent influence of misinformation source to central nodes in OSNs under IC model. In essence, all of these situations are equivalent to removing set of nodes to achieve their purposes. Nevertheless, the above studies were conducted on IC, LT. They are probability models, in which the exact computation the influence from a set nodes in network is #P-hard. This leads to the proposed algorithms have high complexity. Besides, they have not mentioned for deadline in limiting the influence of the known epidemic sources.

In fact, information is spread from one user to another through each hop of the propagation on OSNs. The earlier we can prevent the dispersion of misinformation, the smaller the damage cause. Moreover, recent studies have revealed that the propagation often fades quickly within only few hops from the sources. For example, Cha et al. show that the length of typical chain is less than four [2] and the influence on social networking occurs at the level of direct friends [14].

Motivated by the phenomenon, we define the problem of *Limiting the Spread of Epidemics* (LSE), which seeks to a set of nodes A

SoICT 17, December 2017, Nha Trang, Vietnam

ACM ISBN 978-1-4503-5328-1/17/12…$15.00
https://doi.org/10.1145/3155133.3155157

has size at most k to remove from network to maximum the number of saved nodes within the time constraints. The new aspects in our model are: the influence is limited to within $d \geq 1$ propagation hops from the epidemic sources, and we formulate the problem in a *deterministic difussion model* because the fact that user personal information may be obtained by surveys and data mining techniques [20]. The main challenges of solving the problem came from, it is proven to be NP-hard and it is NP-hard to approximate the LSE problem with ratio $n^{1-\epsilon}$. In this paper, we investigate LSE problem and develop the solutions. Our contributions are summarized as follows:

- We introduced an information propagation model with a time limit of propagation which extended Deterministic Linear Threshold model call *Time Constriant Deterministic Linear Threshold* (T-DLT) model. In this model, we formulated *Limiting the Spread of Epidemics* (LSE) problem that seeks to a set of nodes A has size at most k to remove from network to maximum the number of saved nodes within the time constraints. We showed that the problem is NP-hard and it can not be approximate with ratio of $n^{1-\epsilon}, 0 < \epsilon < 1$.
- For solutions, we first proposed Greedy algorithm that select the node that has maximum incremental saved nodes. To scalable for large-scale networks, we further proposed an efficient heuristic algorithm called *Fast and effective Limiting Epidemics* (FLE).
- Experiments were performed on real-world social traces of Gnutella, Wikipedia Vote, Amazon and Google Web datasets show that our purposed algorithms outperform other methods in terms of maximizing the objective function for each network. FLE performance nearly close to Greedy algorithm and scales up to networks of millions of links.

Outline of the paper. The rest of the paper is organized as follows. We first discuss the related work in Section 2 and the propagation models, problem definition in Section 3. Section 4 introduces some hardness and complexity results. Section 5 presents our proposed algorithms. The experimental results on several datasets are in Section 6. Finally, we give some implications for future work and conclusion in Section 7.

2 RELATED WORK

Information diffusion models are the basis for the study epidemics on OSNs. Domingos and Richardson were first studied information and influence propagation problem on social networks [6]. They designed strategies to spread information and analyzed them based on a data mining technique. Kempe et al. [8] formulated *Influence Maximization* (IM) problem based on two probabilistic models *Independent Cascade* (IC) and *Linear Threshold* (LT) model and proposed a greedy algorithm with a ratio of $(1 - 1/e)$. This issue has become a hot topic in the recent years. Later, many works about designed efficiency algorithms for IM problem and its variants [3, 4, 7, 19].

To decontaminate the misinformation propagation, some authors suggested selecting a set of nodes to inital good information and spread it on the network [1, 22, 27]. However, in reality, once the nodes were infected by misinformation, it is difficult to decontaminate them. The other studies seek to limit the effect of known epidemics. Zhang et al. [26] proposed τ-MP problem that prevents propagation from given misinformation sources to center nodes with guaranteed threshold $\tau \in [0, 1]$ by placing 'monitors'. Note that, placing the monitors at the set of nodes is equivalent to removing them from network. Under IC model, the problem was showed #P-hard, they designed greedy strategies based on the cut-sets-2 technique. However, they did not show the runtime of the algorithm in their experiments. Given a set of notes were infected in a network, Zhang et al. [28] proposed a vaccination strategy for the k nodes so that the number of infected nodes after propagation under IC mode is minimal, vaccination at a node means that the node is immune to source of the epidmics. They given some heuristic algorithms to find solutions but they were not theoretically guaranteed.

In general, the above studies were aimed to preventing epidemics outbreaks by removing nodes from the network in probability diffusion models (IC and LT). However, the time constraint or deadline was not cosidered. In fact, restricting the disease is very difficult when it has erupted and restricting them sooner, the higher the efficiency. In addition, algorithms for IC and LT have high compexity because the the fact that exact computation of influence from soures is difficult #P-Hard [3, 4]. Therefore, our works differrent from their works when we find the set of nodes to remove from the network considering two new factors (1) in a deterministic difussion model, and (2) consider deadline of propagation in this model.

3 MODEL AND PROBLEM DEFINITION

3.1 Difussion Model

The most well known models are LT and IC model model [8]. However, they are propability difussion models and exactly calculating the influence from a given set of node in network is #P-hard [3, 4] (a #P problem is at least as hard as the corresponding NP problem). In this subsection, we describe our extension to the DLT model [20] that incorporates time-constraint diffusion processes, which we call T-DLT. This model can overcome the above disadvantage of probability models, in which computation of influence from a set of nodes can be done in polynomial time. The details of T-DLT model are described as follows.

Graph notations. Let $G = (V, E, w)$ represents a social network with a node set V and a directed edge set E, with $|V| = n$ and $|E| = m$. A node represents a user in the social network, while an edge $e = (u, v)$ in E represents the relationship between users u and v respectively. We denote in-coming and out-coming neighbor nodes of u are $N_+(u)$ and $N_-(u)$, $d_+(u)$ and $d_-(u)$ are in-degree and out-degree of node u. Let $I \subset V$ is the set of *inital infected* nodes. Each directed edge (u, v) has a weight $w(u, v)$ which denotes how much the node u is influenced by its neighbor v satisfy $\sum_{u \in N_+(v)} w(u, v) \leq 1$.

The states of nodes. The process of spreading epidemic from the set I to other nodes on OSNs develops through discrete time steps $t = 1, 2, .., d$. Each node $v \in V$ has two possible states are *infected* and *healthy*.

Infected threshold. Each node v has a pre-fixed thresshold $\theta_v \in [0, 1]$. This represents the weighted fraction of in-coming neighbors of v that must become infected in order for v to become infected.

Difussion process. The diffusion process unfold deterministically is discrete steps (propagation hops) $t = 0, 1, 2, .., d$. Let I_t is set of infected nodes after step t, the propagation process is simulated as follows:

- At round $t = 0$, all nodes in I are infected, i.e, $I_0 = I$.
- At round $t \geq 1$ a healthy node v is infected if the weighted number of its in-coming infected neighbors reaches its infected threshold, i.e.,

$$\sum_{\texttt{in-coming infected neighbor } u} w(u, v) \geq \theta_v \qquad (1)$$

- Once a node becomes infected, it remains status in all subsequent rounds. The influence propagation stops when $t = d$.

In LT model [8], the threshold values θ_v are assigned uniformly at random from the interval $[0, 1]$ and are updated during the spread process reflects our lack of knowledge of the users' internal thresholds. Therefore, this model belongs to the stochastic diffusion model. Different from the LT model, in T-DLT model the infected threshold θ_v of each nodes is given. In this case, the value of threshold θ can be obtained by surveys or data mining techniques same as deterministic linear threshold model (DLT) model [20].

Compare T-DLT and DLT models, the new aspect in our model is that the influence is limited to the nodes from the inital infected nodes within d hop of propagations, for $d \geq 1$. In other words, the difference of T-DLT is the propagation process ends after the d propagation hops. The DLT model is a special case of T-DLT with $d = |V|$.

3.2 Problem Definition

Denoted by $f_d(I)$ is the set of infected nodes after d hops on $G = (V, E)$ in T-DLT model, $f_d(I, A)$ is the set of infected nodes after remove set of nodes $A \subseteq V \setminus I$ from G (i.e, the number of infected nodes in the residual network). Now, the number of saved nodes after removing A is:

$$h_d(A) = |f_d(I, \emptyset)| - |f_d(I, A)| \qquad (2)$$

In T-DLT model, we formulate a combinatorial optimization is *Limiting the Spread of Epidemics* (LSE) problem which aims to seek a set of size at most k nodes to remove from the network so that the number of saved nodes is maximal.

Definition 3.1 (LSE problem). Given an undirected graph $G = (V, E)$ represents a social network under T-DLT model. A set inital infected nodes $I \subset V$ and d is hop of constraint and a budged $k > 0$. Find the set k nodes $A \subseteq V \setminus I$ to remove from network so that the number of saved nodes after d rounds $h_d(A)$ is maximal.

Denote $G_d = (V_d, E_d)$ is a subgraph of $G = (V, E)$ with V_d is the set of nodes whose distance between to each node in I at most d and E_d is the set of edges on the paths from each node in I have distance at most d and $n_d = |V_d|, m_d = |E_d|$. We see that the spread of epidemics only occurs on G_d. Hence, to simplify, instead of considering all node in G, we only find the solution in G_d.

4 COMPLEXITY AND INAPPROXIMATION OF LSE PROBLEM

In this section, we show the NP-hardness of LSE problem by reducing it from the Set Cover problem. We further prove the inapproximability of LSE which is NP-hard to be approximated within a ratio of $n^{1-\epsilon}, (0 < \epsilon < 1)$.

Theorem 4.1. LSE *is NP-hard in T-DLT model.*

Proof. To prove that LSE is NP-hard, we reduce the known NP-complete is decision of *Set Cover* problem to it.

Set Cover (SC) problem. Given a positive integer t, a universal set $\mathcal{U} = \{e_1, e_2, ..., e_M\}$ and a collection subsets $\mathcal{S} = \{S_1, S_2, ..., S_N\}$, we can assume that $t < M < N$. The Set Cover problem asks to whether or not there are t subsets whose union is $\mathcal{U}$?

The SC problem was proved NP-hard even when the sizes of subsets are bounded by 3 and each element appears in exactly two subsets [5]. To reduce SC problem to LSE, we first construct an instance $\mathcal{I}_{LSE}$ of LSE from an instance $\mathcal{I}_{SC}$ of SC problem. We then we show that if $\mathcal{I}_{SC}$ has a solution S of size t, $\mathcal{I}_{LSE}$ has a solution $A, |A| \leq k$ such that $h_d(A) \geq k + M$ and otherwise.

Construction. Given an instance $\mathcal{I}_{SC} = \{\mathcal{U}, \mathcal{S}, t\}$ of the SC problem, we construct an instance $\mathcal{I}_{LSE} = \{G, I, d, k\}$ of LSE problem as follows (fig. 1).

- *The set of nodes and edges*: for each $S_i \in \mathcal{S}$, we construct an inital infected node $s_i \in I$ and a vertex u_i. We add a directed edge (s_i, u_i). For each element $e_j \in \mathcal{U}$ we add a node v_j and add a directed edge (u_i, v_j) for each $e_j \in S_i$. For convenience, we let $B = \{u_1, u_2, \ldots, u_N\}, C = \{v_1, v_2, \ldots, v_M\}$.
- *Infected thresholds and weights*: we assign $w(s_i, u_i) = 1, w(u_i, v_j) = \frac{1}{d_+(v_j)}, \theta_{u_i} = \theta_{v_j} = 1$.
- Finally, we set $k = t, d = 2$.

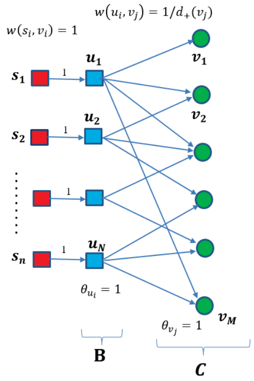


Figure 1: Reduce from MC problem to LSE problem.

Analysis. For the construction, we found that $|f_d(I, \emptyset)| = M + N$. If any in-coming neighbour node $u_i \in B$ of $v_j \in C$ is healthy nodes then:

$$\sum_{\texttt{in-coming infected neighbour } u} w(u, v_j) \leq 1 - 1/d_+(v_j) < 1 = \theta_{v_j}$$

So v_j is healthy node (not infected). Otherwise, all nodes in C are healthy nodes.

($\longrightarrow$) Suppose $\mathcal{S}'$ is solution of instance $\mathcal{I}_{SC}$ that means $|\mathcal{S}'| = t = k$ and it cover t elements of $\mathcal{U}$. If we chose set A contains node u_i corresponding to $S_i \in \mathcal{S}'$, (i.e, $A = \{u_i | S_i \in \mathcal{S}'\}$), every node in $v_j \in B$ adjacent to at least one node in A. By above analysis, all nodes in C are healthy nodes. We have $h_d(A) = t + M = k + M$.

($\longleftarrow$) Otherwise, if $\mathcal{I}_{LSE}$ has the solution $A, |A| \leq k$ with $h_d(I, A) \geq k + M$. If A contain t_1 $(1 \leq t_1 \leq k)$ nodes in C, $h_d(I, A) \leq k - t_1 + M <$

$k + M$. So A doesn't contain any node in C. Hence, $A \subset B$. Combine with $h_d(I, A) \geq k + M$, all nodes in C are healthy nodes. Therefore, every node $u_i \in A$ adjacent to at least one node in C (above analysis). Now with our construction we see that, $\mathcal{S}' = \{S_i | u_i \in A\}$ is the solution of $\mathcal{I}_1$. This complete the proof. □

Base on modifying the above reduction and gap-introduction reduction [24], we further prove the inapproximability of the LSE problem.

Theorem 4.2. *It is NP-hard to approximate the LSE problem with ratio $n^{1-\epsilon}$ in T-DLT model for any constant $0 < \epsilon < 1$.*

Proof. To prove this result, we user the gap-introduction reduction in [24] to prove the inapproximability of the LSE. Using a polynomial time reduction from Set Cover to LSE, we show that if there exists a polynomial time algorithm that approximates the later problem within $n^{1-\epsilon}$, then there exists a polynomial time algorithm to solve the former problem.
Contruction. Given an instance of the SC problem (as theorem 4.1) $\mathcal{I}_{SC} = (\mathcal{U}, \mathcal{S}, t)$, we construct an instance $\mathcal{I}_{LSE} = (G, I, d, k)$ by reusing the construction in theorem 4.1 and add some vertices and edges as illustrated in fig. 2.

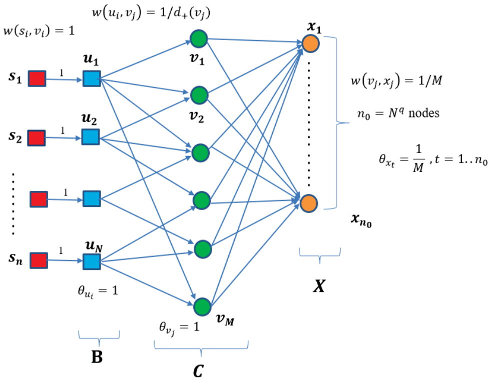


Figure 2: Reduce from Set Cover to LSE.

- *The set of nodes and edges*: For each node $v_j \in C$ we add $n_0 = N^q$ more nodes $X = \{x_1, x_2, ..., x_{n_0}\}$ for an arbitrarily large constant q and add directed edge $(v_j, x_l), l = 1..n_0$.
- *Infected thresholds and weights*: we assign $w(v_j, x_l) = \frac{1}{M}$; $\theta_{x_l} = \frac{1}{M}, l = 1..n_0$.

Assume that $\mathcal{I}_{SC}$ has a set cover $\mathcal{S}'$ of size t, we select set $A = \{u_i | S_i \in \mathcal{S}'\}$. By the alanysis in theorem 4.1, all nodes in C are healthy nodes. This leads to all nodes in X are also healthy nodes. For this construction, we have: $|f_d(I, \emptyset)| = N + M + N^q$, $|f_d(I, A)| = N - k$ so:

$$h(A) = |f_d(I, \emptyset)| - |f_d(I, A)| = M + N^q + k > N^q$$

In the case $\mathcal{I}_{SC}$ has no set cover of size t, it has at least a vertex v_j infected, This lead to all vertices in X are infected. Therefore $|f_d(I, A)| > N - k + N^q$ inferred $h(A) = M + k < 2N$.
Now, supposed that we have polynomial algorithm $\mathcal{A}$ which approximates LSE problem within ratio $n^{1-\epsilon}$, we show that the solution of SC problem can be found in polynomial time. For any instance $\mathcal{I}_{SC}$ we construct an instance $\mathcal{I}_{LSE}$ as above construction in polynomial time function of m and n.
In the case $\mathcal{I}_{SC}$ has a set cover size t, by our construction, the optimal solution A_{opt} of $\mathcal{I}_{LSE}$ has $h(A_{opt}) = N^q + M + k$. The algorithm $\mathcal{A}$ approximate the optimal solution within ratio $n^{1-\epsilon}$ ($n = 2N + M + N^q$ is number of nodes of input graph), so it can find a solution $\mathcal{A}(\mathcal{I}_{LSE})$.

$$\begin{aligned} h(\mathcal{A}(\mathcal{I}_{LSE})) &\geq \frac{1}{n^{1-\epsilon}} h(A_{opt}) > \frac{1}{n^{1-\epsilon}}(M + N^q + k) \\ &= \frac{n^\epsilon}{n}(M + N^q + k) > \frac{(N^q + 2N + M)^\epsilon}{N^q + 2N + M} N^q \\ &> \frac{N^{q.\epsilon}}{4N^q} N^q = \frac{1}{4} N^{q.\epsilon} \end{aligned}$$

We choose q large enough, so that $q > \frac{\ln(8N)}{\epsilon \ln N}$ then

$$h(\mathcal{A}(\mathcal{I}_{LSE})) > 2N$$

On the other hand, if $\mathcal{I}_{SC}$ has no set cover of size t, then the optimal set A_{opt} of $\mathcal{I}_{LSE}$ has

$$h(\mathcal{A}(\mathcal{I}_{LSE})) < 2N$$

This implies the $\mathcal{I}_{SC}$ has a set cover of size t if only if $h(\mathcal{A}(\mathcal{I}_{LSE})) < 2N$. Hence, we can use $\mathcal{A}$ to decide the SC problem in polynomial time i.e., $P = NP$. This contradicts the hypothesis that $P \neq NP$. □

5 PROPOSED ALGORITHMS

5.1 Greedy Algorithm

On the problems of information propagation, one can use Greedy method to find a good enough solution. We first introduced a straightforward greedy algorithm in algorithm 1. The algorithm sequentially selects a node u into the selected set A that maximizes the *incremental saved nodes*:

$$\delta(A, u) = |h_d(I, A \cup \{u\})| - |h_d(A)| = |f_d(I, A)| - |f_d(I, A \cup \{u\})| \tag{3}$$

Lemma 5.1. *Given graph $G = (V, E)$ and number of propagation hops d, the objective function $h_d(I, A)$ can be computed withm $O(m_d + n_d)$*

Proof. The number of infected nodes after removing any set of nodes A $f_d(I, A)$ can be computed by using a Breadth-First Search (BFS) in graph G_d. The time taken is $O(m_d + n_d)$. By equation 2, we infer $h_d(I, A)$ can be done in $O(m_d + n_d) + O(m_d + n_d) = O(m_d + n_d)$. □

Theorem 5.2. *The complexity of Greedy Algorithm is $O(kn_d(m_d + n_d))$.*

Proof. The two loops in line 2 and line 4 contribute a factor kn_d to the time complexity. From Lemma 5.1 and equation 3, computing the number of incremental saved nodes (line 6) can be done in $O(m_d + n_d)$, the total time complexity of the algorithm is $O(kn_d(m_d + n_d)$. □

Based on theorem 1, in the worst case, k can be as large as n, the algorithm 1 can take $O(n_d^2(m_d + n_d))$. It is intractable for even medium-size social networks. To overcome this, we introduce a

Algorithm 1: Greedy algorithm

```
Data: Graph $G = (V, E, \theta)$, inital infected nodes $I$, propagation hop $d$
Result: The selected nodes $A$
$A \leftarrow \emptyset$;
for $i = 1$ to $k$ do
    $\delta_{max} \leftarrow 0$
    foreach $v \in V \setminus A$ do
        if $\delta(A, u) > \delta_{max}$ then
            $\delta_{max} \leftarrow \delta(A, u)$
            $u_{max} \leftarrow u$
        end
    end
    if $\delta_{max} = 0$ then
        return $S$
    else
        $A \leftarrow A \cup \{u_{max}\}$
    end
end
return $A$;
```

Algorithm 2: Fast Limit Epidemics (FLE) algorithm

```
    Data: Graph $G = (V, E, \theta)$, set of initial infected nodes $I$, propagation hop $d$.
    Result: set of nodes $A$
1.  $S \leftarrow \emptyset$;
2.  Calculate $G' \leftarrow G_d(I)$; $U \leftarrow V_d$
3.  for $i = 1$ to $k$ do
4.      Calculate $\alpha(u), \beta(u)$ on $G'$ (Algorithm 3)
5.      $u_{max} \leftarrow 0$
6.      if $\beta(v) = 0, \forall v \in U$ then
7.          $u_{max} \leftarrow \arg\max_{v \in U} \alpha(v)$
8.      else
9.          $u_{max} \leftarrow \arg\max_{v \in U} \beta(v)$
10.     end
11.     $S \leftarrow S \cup \{u_{max}\}$
12.     $U \leftarrow U \setminus \{u_{max}\}$
13.     Remove $u_{max}$ and all edges that adjacent with $u_{max}$ from $G'$
14. end
15. return $S$;
```

new heuristic algorithm based on evaluating the role of spreading disease of each node. The new algortihm can find fast solutions for medium and large-size social networks while still ensuring performance against greedy algorithm in our experimental study.

5.2 Fast And Effective Limiting Epidemics Algorithm

In the previous section, we introduced the greedy algorithm. However, the time complexity is quite high, due to the fact that calculating the number of incremental saved nodes usually results in a long running time. From this point of view, in this section we design a heuristic algortihms with low time complexity and highly expected. We begin this section by defining the terminologies used in our proposed solution as follows.

- $t(u)$: the number smallest hop when u change from healthy to infected state.
- $a_+(u) = \sum_{I_{t(u)-1} \cap N_+(u)} w(u, v)$ is total weight of of in-edges from in-coming infected neighbors before hop $t(u)$.
- $\alpha(u) = \sum_{v \in \bigcup_{i=t(u)+1}^{d} I_i \cap N_-(u)} w(v, u), i = t(u) + 1, \ldots, d$ is total weight of out-edges from u to out-coming infected neighbor nodes v at hop i.
- $\beta(u)$: the number of out-neighbors nodes, which change from infected to healthy state after removing node u from graph.

Intuitively, the number of incremental saved nodes $\delta(A, u)$ can be approximated by $\beta(u)$. Besides, in order to increase the efficiency of proposed algorithm, we combine $\alpha(u)$ with $\beta(u)$ to measure the role of epidemic propagation of node u. The main idea of the algorithm is to select the node in each step based on the evaluation of the α and β functions. Initially, we initialized the selected nodes $A = \emptyset$ and set candidate nodes U is equal to V_d. In each step, we select the node u so that $\beta(u)$ is maximal on residual graph. In the case, all candidate nodes have same $\beta(.)$ value, which is equal to zero, we choose node with the maximum $\alpha(.)$ value. The algorithm is depicted in Algorithm 2.

The difficulty in performing algorithm 2 comes from the calculation of $\alpha(.), \beta(.)$ function. In order to do this, we first used the idea of Breath First Search (BFS) to calculate $f_d(I)$ and incremental update $a_+(.)$ function. Then, we used it to calculate the $\alpha(.), \beta(.)$ according to their definition. The details of the algorithm are shown in Algorithm 3 pseudo-code.

Theorem 5.3. *The complexity of algorithm 2 is* $O(k(m_d + n_d))$

Proof. To analyze the complexity of this algorithm, we first needed to evaluate the complexity of algorithm 3. On algorithm 3, we easily see that, calculating $f_d(I)$ (line 5-19) is similar to the mechanism of BFS. Therefore, the complexity of this work is $O(m_d + n_d)$. For the calculating $\alpha(u)$ and $\beta(u)$ pharse (line 21-31), this works need to visit all nodes in $f_d(I)$ and the set out-coming edge neighbors, this work it take at most $O(m_d + n_d)$. Therefore the complexity of algorithm 3 is $O(m_d + n_d)$. In algorithm 2, for each the main loop (line 3-14), choosing u such that $\beta(u)$ and $\alpha(u)$ maximal can be done in linear time. Therefore, this task has complexity $O(m_d + n_d) + O(n_d) = O(m_d + n_d)$. In summary, the complexity of algorithm 2 is $O(k(m_d + n_d))$. □

6 EXPERIMENTS

In this section, we perform experiments on OSNs to show the efficiency of our proposed algorithms in comparison to some baseline mdethods which were used for some problems of information propagations [8, 21, 26, 28]. We compare on three aspects: the *solution quality*, the *scalability* and impact of propagation hops d for various real social network datasets. The baseline algorithms used include:

- Random: Randomly select nodes within number of nodes k among the $N_d(S)$.

Algorithm 3: $f_d(I), a_+(u), t(u), \alpha(u), \beta(u), \forall u \in V_d$ on $G = (V, E)$

```
Data: Graph $G = (V, E, \theta)$, set of infected nodes $I$, propagation hop $d$
Result: $f_d(I), a_+(u), t(u), \alpha(u), \beta(u), \forall u \in G$
$t(u) \leftarrow +\infty, a_+(u) \leftarrow 0, \alpha(u) \leftarrow 0, \beta(u) \leftarrow 0, \forall u \in V_d$
$t(s) \leftarrow 0, \forall s \in I$
Queue $Q \leftarrow \emptyset$
\\ Calculate $a_+(u), t(u), f_d(I)$
while $Q \neq \emptyset$ do
    $u \leftarrow Q.pop()$
    $f_d(I) \leftarrow u$
    if $t(u) < d$ then
        foreach $v \in N_-(u)$ do
            if $t(u) < t(v)$ then
                $a_+(v) = a_+(v) + w(u, v)$
                if $(a_+(v) \geq \theta_v)$ and $(t(u) + 1 < t(v))$ then
                    $t(v) \leftarrow t(u) + 1$
                    $Q.push(v)$
                end
            end
        end
    end
end
\\ Calculate $\alpha(u), \beta(u)$
foreach $u \in f_d(I)$ do
    $\alpha(u) \leftarrow 0; \beta(u) \leftarrow 0$
    foreach $v \in N_-(u) \cap f_d(I)$ do
        if $t(u) < t(v)$ then
            $\alpha(u) \leftarrow \alpha(u) + w(u, v)$
            if $a_+(v) - w(u, v) < \theta_v$ then
                $\beta(u) \leftarrow \beta(u) + 1$
            end
        end
    end
end
return $A$;
```

- Max Degree (Maxdeg): The heuristic algorithm base centrality measure. We select nodes with highest degree among the V_d and we keep on adding highest-degree nodes until the number of selected nodes is equal to k.

We conducted the experiments using a Intel(R) Core(TM) i5-6200U CPU @ 2.30 GHz (up to 2.40 GHz) machine with 4.0 GB of 1066Mhz main memory, and implemented the algorithms in C++.

6.1 Datasets

We ran our experiments on multiple real datasets. In addition to trying to pick datasets of various sizes, we also chose from different domains, in which the LSE problem is especially applicable. Table 1 shows datasets we used.

Gnutella. The snapshot of the Gnutella peer-to-peer file sharing network in August 2002. Nodes represent hosts in the Gnutella network topology and edges represent connections between the Gnutella hosts [17]. It contains 20,777 links among 6,301 hosts.

Table 1: Datasets

Network	#Nodes	#Edges	Type	Avg. Deg
Gnutella [17]	6,301	20,777	Directed	3.29
Wikipedia vote [16]	7,115	103,689	Directed	14.57
Amazon [15]	262,111	1,234,877	Directed	4.71
Google Web [18]	875,713	5,105,039	Directed	5.83

Wiki Vote. The network contains all the Wikipedia voting data from the inception of Wikipedia till January 2008. Nodes in the network represent wikipedia users and a directed edge from node i to node j represents that user i voted on user j [16].

Amazon. Network was collected by crawling Amazon website. It is based on Customers Who Bought This Item Also Bought feature of the Amazon website. If a product i is frequently co-purchased with product j, the graph contains a directed edge from i to j. The data was collected in March 02 2003 [15].

Google Web. Nodes represent web pages and directed edges represent hyperlinks between them. The data was released in 2002 by Google as a part of Google Programming Contest [18].

6.2 Experimental Settings

The edge weights. They assign the weights of edges according to previous studies [4, 7–9]. Accordingly, the weight of the edge (u, v) is calculated as follows:

$$w(u, v) = \frac{1}{d_+(v)} \tag{4}$$

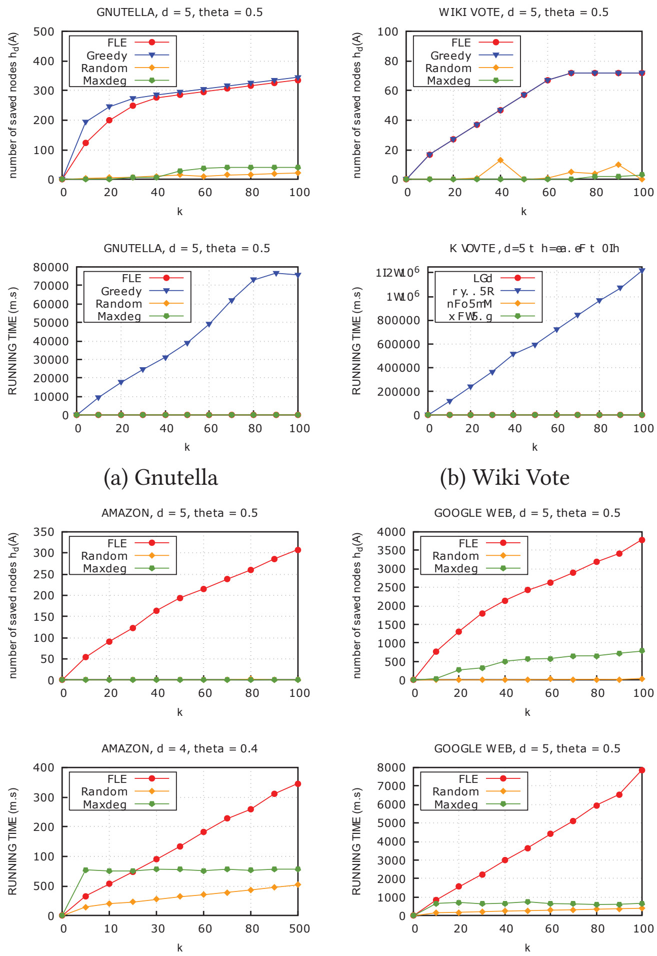
where $d_+(v)$ denotes the in-degree of node v.

Infected thresshold and propagation hops. In fact, we see that with each user, the more infected they are, the more likely they are infected. Therefore, we choose the infected threshold in the set $\{0.1; 0.2; 0.3; 0.4; 0.5; 0.6; 0.7; 0.8; 0.9\}$. Cha et al. [2] showed that information mostly propagates within 2 to 5 hops so we choose $d = 2, 3, 4, 5, 6$.

In all experiments, we choose we choose a set inital infected nodes that contain 1% of set of nodes for each network.

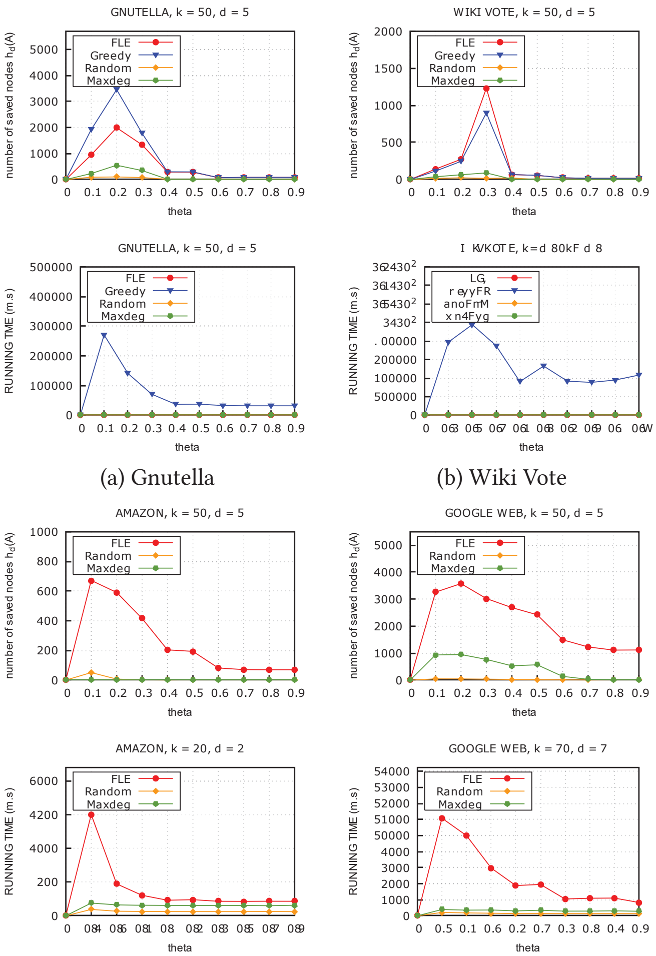
6.3 Experiment Results

Solution Quality. We evaluate the performance of algorithms in three cases: (1) the number of selected nodes k changes from 10 to 100 and $d = 5, \theta = 0.5$ (fig. 3), (2) the threshold θ changes, d and $k = 50$ are fixed (fig. 4), (3) the number of propagation hops d changes and θ and k are fixed (fig. 5). Under all circumstances, Greedy and FLE yielded more desirable results than the remaining algorithms. The number of saved nodes is up to 48.5 times higher in comparison with Maxdeg and Random algorithm in Gnutella and Wiki Vote. Comparing Greedy and FLE, we found that, Greedy has 1.02 to 1.5 time better performance in Gnutella network. However, the gap between them is narrowed when k, θ increase. Specifically, when $k \geq 50, \theta \geq 0.4$, their performance is almost the same. Figure 3 (b) and fig. 4 (b), fig. 5 (a) reveal that the Greedy and FLE achieve nearly the same level of performance in terms of number of saved nodes for Wiki Vote dataset. While Greedy cannot finish on Amazon



(a) Gnutella (b) Wiki Vote

(c) Amazone (d) Google Web



(a) Gnutella (b) Wiki Vote

(c) Amazone (d) Google Web

Figure 3: Comparising number of saved nodes and time running of algorithms FLE, Greedy, Maxdeg, Random **on LSE prblem when k varies and $d = 5, \theta = 0.5$.**

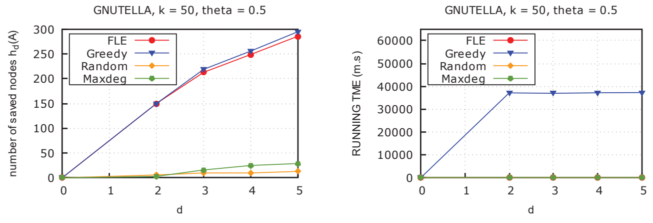
Figure 4: Comparising number of saved nodes and time running of algorithms FLE, Greedy, Maxdeg, Random **on LSE prblem when θ varies and $k = 50, d = 5$.**

and Google Web after 12h and was forced to terminate, FLE is still better with the remaining algorithms.

Scalability. The running times of all algorithms is also presented in fig. 3, 4, 5. As expected, the running time for Greedy is extremely higher than other algorithms, taking up to 4.5 min for Gnutella and 20.2 min for Wkiki Vote. FLE is 4,820 to 6,879 times faster than Greedy in Gnutella and FLE is 5,839 to 14,490 times faster than Greedy in Wiki Vote. This is because Greedy has a $O(n_d k(m_d+n_d))$ complexity while FLE takes in $O(k(m_d + n_d))$. For Amazon and Google Web, Greedy algorithm cannot finish within 12h as above mentioned while FLE finish in 0.45 s and 7.8 s. This proves that FLE still works well for million-scale networks.

Impact of d. We also explore the behavior of different methods when the number of propagation hops d is varied from 2 to 6 in Gnutella. The results are shown in fig. 5 (the result for other d values are consistent with that of $d = 5$). With Greedy and FLE, the number of saved nodes increases when d increases. Specifically, it increased rapidly when $d = 2; 3$ and slowler when $d = 4; 5$. This proves that to limit the epidemics, we need to stop as soon as possible (i.e, d small).

Impact of θ. We investigate the effects of θ, when θ changes, $kandd$ are fixed. We consider $d = 5, k = 50$ the results as show in fig. 4; however, the result is quite similar for other case. For Amazon and Google Web, the number of saved nodes decreasing when θ continues decreasing. For Gnutella and Wiki Vote the number of saved nodes increasing when θ increases from 0.10 to 0.30 and decreasing when θ increases from 0.30 to 0.90. In general, we can conclude the fact that the larger the θ, the spread of the epidemics more difficult. From fig. 4 we can see that for Gnutella and Wiki Vote data, Greedy and FLE give large number of saved nodes compared to the other methods when θ is varied from 0.10 to 0.90. This is, again, in conformity with the earlier results of fig. 3 and supports the superiority of the proposed algorithms.



(a) (b)

Figure 5: The number of saved nodes $h_d(A)$ when d varies and $k = 50, \theta = 0.5$ on Gnutella

7 CONCLUSION

In this paper, we investigate a new NP-hard problem of seeking a set of nodes has size at most k to remove from network to maximum the number of saved nodes within the time constraints for given infected nodes. We prove the inapproximability result and provide the greedy method to solve the problem. In addition, due to the high runtime complexity of this greedy algorithm, a much more efficient heuristic algorithm called FLE is also proposed. The result of the experiment shows that the suggested algorithms are better than the baseline algorithms Maxdeg and Random. Besides, the performance of FLE is nearly close to Greedy algorithm and scales up to networks of millions of links. In the future, we will improve FLE algorithm to achieve performance closer to the optimal solution.

REFERENCES

[1] Ceren Budak, Divyakant Agrawal, and Amr El Abbadi. 2011. Limiting the spread of misinformation in social networks. In *Proceedings of the 20th international conference on World wide web*. Hyderabad, India, 665–674. https://doi.org/10.1145/1963405.1963499

[2] M. Cha, A. Mislove, and K. P. Gummadi. 2009. A measurement-driven analysis of information propagation in the Flickr social network. In *Proceedings of the 18th international conference on World wide web (2009)*. ACM, Madrid, Spain, 721âĂŞ730. https://doi.org/10.1145/1526709.1526806

[3] W. Chen, C. Wang, and Y. Wang. 2010. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. Washington, USA, 1029–1038. https://doi.org/10.1145/956750.956769

[4] W. Chen, C. Wang, and Y. Wang. 2010. Scalable Influence Maximization in Social Networks under the Linear Threshold Model. In *Proceedings of the 2010 IEEE International Conference on Data Mining*. Washington, DC, USA, 88–97. https://doi.org/10.1145/956750.956769

[5] Miroslav Chlebik and Janka Chlebikova. 2006. Complexity of approximating bounded variants of optimization problems. *Theoretical Computer Science* 354, 3 (April 2006), 320–338. https://doi.org/10.1016/j.tcs.2005.11.029

[6] P. Domingos and M. Richardson. 2001. Mining the network value of customers. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. San Francisco, CA, USA, 57–66. https://doi.org/10.1145/502512.502525

[7] Amit Goyal, Wei Lu, and Laks V.S. Lakshmanan. 2011. SIMPATH: An Efficient Algorithm for Influence Maximization under the Linear Threshold Model. In *Proceeding IEEE 11th International Conference on Data Mining*. Vancouver, Canada, 211–220. https://doi.org/10.1109/ICDM.2011.132

[8] D. Kempe, J. Kleinberg, and E. Tardos. 2003. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. Washington, DC, USA, 137–146. https://doi.org/10.1145/956750.956769

[9] Elias Boutros Khalil, Bistra Dilkina, and Le Song. 2014. Scalable diffusion-aware optimization of network topology. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. New York, NY, USA, 1226–1235. https://doi.org/10.1145/2623330.2623704

[10] M. Kimura, K. Saito, , and H. Motoda. 2008. Solving the contamination minimization problem on networks for the linear threshold model. In *PRICAI 2008: Trends in Artificial Intelligence*, Tu-Bao Ho and Zhi-Hua Zhou (Eds.). Hanoi, Vietnam, 977–984. https://doi.org/10.1007/978-3-540-89197-0_94

[11] M. Kimura, K. Saito, , and H. Motoda. 2009. Blocking links to minimize contamination spread in a social network. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 3, 2 (April 2009), 795–825. https://doi.org/10.1145/1514888.1514892

[12] Ivana KottasovÃą. 2017. Facebook targets 30,000 fake accounts in France. http://money.cnn.com/2017/04/14/media/facebook-fake-news-france-election/index.html

[13] Sejeong Kwon, Meeyoung Cha, Kyomin Jung, Wei Chen, and Yajun Wang. 2013. Prominent features of rumor propagation in online social media. In *Proceeding of IEEE 13th International Conference on Data Mining*, Hui Xiong, George Karypis, Bhavani Thuraisingham, Diane Cook, and Xindong Wu (Eds.). Institute of Electrical and Electronics Engineers (IEEE), Dallas, Texas, US, 1103–1109. https://doi.org/10.1109/ICDM.2013.61

[14] J. Leskovec, L. A. Adamic, and B. A. Huberman. 2007. The dynamics of viral marketing. *ACM Transactions on the Web* 1, 1 (May 2007). https://doi.org/10.1145/1232722.1232727

[15] J. Leskovec, L. A. Adamic, and B. A. Huberman. 2007. The dynamics of viral marketing. *ACM Transactions on the Web* 1, 1 (May 2007).

[16] J. Leskovec, D. Huttenlocher, and J. Kleinberg. 2010. Predicting Positive and Negative Links in Online Social Networks. In *Proceedings of the 19th international conference on World wide web*. Raleigh, North Carolina, USA, 641–650. https://doi.org/10.1145/1772690.1772756

[17] J. Leskovec, J. Kleinberg, and C. Faloutsos. 2007. Graph Evolution: Densification and Shrinking Diameters. *ACM Transactions on Knowledge Discovery from Data* 1, 1 (March 2007), 320âĂŞ338. https://doi.org/10.1145/1217299.1217301

[18] J. Leskovec, K. Lang, A. Dasgupta, and M. Mahoney. 2009. Community Structure in Large Networks: Natural Cluster Sizes and the Absence of Large Well-Defined Clusters. *Internet Mathematics* 1, 1 (2009), 29–123.

[19] B. Liu, G. Cong, D. Xu, and Y.Zheng. 2012. Time Constrained Influence Maximization in Social Networks. In *Proceeding of IEEE 12th International Conference on Data Mining*, Mohammed J. Zaki, Arno Siebes, Jeffrey Xu Yu, Bart Goethals, Geoff Webb, and Xindong Wu (Eds.). Brussels, Belgium, 439–448. https://doi.org/10.1109/ICDM.2012.158

[20] Zaixin Lu, Wei Zhang, Weili Wu, Joonmo Kim, and Bin Fu. 2011. The complexity of influence maximization problem in the deterministic linear threshold model. *Journal of Combinatorial Optimization* 24, 3 (April 2011), 374–378. https://doi.org/10.1007/s10878-011-9393-3

[21] D. T. Nguyen, N. P. Nguyen, and M. T. Thai. 2012. Sources of Misinformation in Online Social Networks: Who to Suspect?. In *MILITARY COMMUNICATIONS CONFERENCE(MILCOM 2012)*. Institute of Electrical and Electronics Engineers (IEEE), Orlando, FL, USA, 1–6. https://doi.org/10.1109/MILCOM.2012.6415780

[22] Nam P. Nguyen, Guanhua Yan, and My T. Thai. 2013. Analysis of misinformation containment in online social networks. *Computer Networks: The International Journal of Computer and Telecommunications Networking* 57, 10 (May 2013). https://doi.org/10.1016/j.comnet.2013.04.002

[23] Vahed Qazvinian, Emily Rosengren, Dragomir R. Radev, and Qiaozhu Mei. 2011. Rumor has it: Identifying misinformation in microblogs. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Reginald N. Smythe and Alexander Noble (Eds.). Association for Computational Linguistics, Edinburgh, Scotland, UK, 1589–1599.

[24] Vijay V. Vazirani. 2001. *Approximation Algorithms*. Springer, Verlag New York, Inc. New York, NY, USA.

[25] Danny Yadron. 2017. Twitter deletes 125,000 Isis accounts and expands anti-terror teams. https://www.theguardian.com/technology/2016/feb/05/twitter-deletes-isis-accounts-terrorism-online

[26] H. Zhang, M. Alim, X. Li, M. T. Thai, and H. Nguyen. 2014. Misinformation in Online Social Networks: Catch Them All with Limited Budget. In *ACM Transactions on Information Systems (TOIS) 2016*. ACM, Shanghai, China, 1719–1728. https://doi.org/10.1145/2661829.2662088

[27] H. Zhang, H. Zhang, X. Li, and M. T. Thai. 2015. Limiting the Spread of Misinformation while Effectively Raising Awareness in Social Networks. In *Proceedings of the 4th International Conference on Computational Social Networks (CSoNet)*. Springer, Beijing, China, 35–47. https://doi.org/10.1007/978-3-319-21786-44

[28] Y. Zhang and B. Prakash. Dava. 2015. Data-Aware Vaccine Allocation Over Large Networks. *ACM Transactions on Knowledge Discovery from Data* 10, 2 (October 2015), 291–301. https://doi.org/10.1145/2803176

[29] Yao Zhang and B. Aditya Prakash. 2014. Scalable Vaccine Distribution in Large Graphs given Uncertain Data. In *Proceeding of the 23rd ACM International Conference on Information and Knowledge Management*. ACM, Shanghai, China, 1719–1728. https://doi.org/10.1145/2661829.2662088