Computer Networks 57 (2013) 2133–2146

Contents lists available at SciVerse ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Analysis of misinformation containment in online social networks

Nam P. Nguyen [a,b,*], Guanhua Yan [b], My T. Thai [a]

[a] CISE Department, University of Florida, Gainesville, USA
[b] Information Sciences (CCS-3), Los Alamos National Laboratory, NM, USA

ARTICLE INFO

Article history:
Received 17 May 2012
Received in revised form 24 January 2013
Accepted 8 April 2013
Available online 20 April 2013

Keywords:
Misinformation containment
Community detection
Information propagation

ABSTRACT

With their blistering expansion in recent years, popular online social sites such as Twitter, Facebook and Bebo, have become not only one of the most effective channels for viral marketing but also the major news sources for many people nowadays. Alongside these promising features, however, comes the threat of misinformation propagation which can lead to undesirable effects. Due to the sheer number of online social network (OSN) users and the highly clustered structures commonly shared by these kinds of networks, there is a substantial challenge to efficiently contain viral spread of misinformation in large-scale social networks. In this paper, we focus on how to limit viral propagation of misinformation in OSNs. Particularly, we study a set of problems, namely the β_T^I-*Node Protectors* problems, which aim to find the smallest set of highly influential nodes from which disseminating good information helps to contain the viral spread of misinformation, initiated from a set of nodes I, within a desired fraction $(1-\beta)$ of the nodes in the entire network in T time steps. For this family of problems, we analyze and present solutions including their inapproximability results, greedy algorithms that provide better lower bounds on the number of selected nodes, and a community-based method for solving these problems. We further conduct a number of experiments on real-world traces, and the empirical results show that our proposed methods outperform existing alternative approaches in finding those important nodes that can help to contain the spread of misinformation effectively.

Published by Elsevier B.V.

1. Introduction

The growing popularity of online social networks, together with their diversity, has drastically changed the landscape of communications and information sharing in cyber-space. Many people have integrated popular online social sites, such as Facebook and Twitter, into their everyday lives and rely on them as one of their major news sources. For example, the news of the hit on Bin Laden first broke out on Twitter long before the US president officially announced it on the public media [1], or the recent event *Occupy of Wall Street* has been spread quickly and widely to a larger population due to its Facebook page [2]. The popularity of OSNs, as a result, is obtained from the convenience as well as efficiency of information dissemination and sharing based on the trust relationships built among their users. Unfortunately, such trust relationships on social networks can possibly be exploited for distributing misinformation or rumors that could potentially cause undesirable effects such as the widespread panic in the general public. For instance, the misinformation of swine flu was observed in Twitter tweets at the outset of the large outbreak in 2009 [3], or the wide spread of the false report that President Obama was killed on the hacked Fox News' Twitter feed in July 2011 [4].

In order for online social networks to serve as a trustworthy channel for disseminating important information,

[*] Corresponding author at: CISE Department, University of Florida, Gainesville, USA. Tel.: +1 7402740545.

E-mail addresses: nanguyen@cise.ufl.edu (N.P. Nguyen), ghyan@lanl.gov (G. Yan), mythai@cise.ufl.edu (M.T. Thai).

1389-1286/$ - see front matter Published by Elsevier B.V.
http://dx.doi.org/10.1016/j.comnet.2013.04.002

it is crucial to have an effective strategy to contain or limit the viral effect of such misinformation. In particular, we aim to find a tight set of users from whom disseminating "good information" minimizes the negative effects of misinformation, or in other words, we want to make sure that most of the network users are aware of the good information by the time the bad information reaches them. Here, the good information can be an authorized announcement to correct the corresponding misinformation. In the above examples, good information could be something simple such as "Swine flu rumor is not correct" or "President Obama is still healthy". However, from whom should the good information be disseminated so that the viral effect of misinformation can be contained in a timely manner, when the infected sources are either known (e.g., hacked Fox News' Twitter feed) or unknown (e.g., tweets about swine flu)?

Due to the sheer number of online social network users and the highly clustered structures commonly shared by these kinds of networks, there is a substantial challenge to efficiently contain viral spread of misinformation in large OSNs. Conventional wisdom mainly focuses on immunization which chooses a set of nodes in the network to immunize in order to disrupt the diffusion process from a graph-theoretic standpoint. In the setting of misinformation containment in OSNs, immunization of certain nodes requires inspecting every message traversing them and stopping those suspected of carrying misinformation. This process itself, however, can be computationally expensive due to the enormous number of messages that spread in a large online social network, e.g., Facebook or Twitter. For example, there were 177 million tweets sent out in a single day on March 11, 2011 [5], and inspecting a tiny URL embedded in a tweet for potential misinformation can be time consuming and inaccurate [6].

Against this backdrop, in this study we consider a scheme that takes a more offensive approach to fight against viral spread of misinformation on social networks. Rather than classifying messages spreading in a network as misinformation or not, this method relies on the similar diffusion mechanism adopted by the misinformation propagation in order to contain it. The key difference, however, is that misinformation often starts from nodes that are less influential and its propagation speed is thus constrained by the trust relationships inherent in the diffusion process from these origins. The containment methods we consider herein, by contrast, aim to find a smallest set of influential people to decontaminate so that the "good" diffusion process starting from them achieves the desirable effect on the spread of misinformation, i.e., the propagation of misinformation is contained within a fraction $(1-\beta)$ of the entire network. We call it the β_T^I*-Node Protector* problem, where β is the desired decontamination threshold, I is the initial infected set (either known or unknown), and T is the time window allowed for decontamination (either constrained or unconstrained). Here, the superscript I or subscript T is shown only if I is known or T is constrained, respectively.

Some attempts on limiting misinformation have been made in earlier works (see Related Work). The most relevant work to our effort is the one suggested by Budak et al. [7], in which the authors formulated this as an optimization problem, proved its NP-hardness, and then provided approximation guarantees for a greedy solution based on the submodularity property. However, the key differences between our work and theirs are: (1) they impose a k-node budget, i.e., the size of the selected set of nodes is constrained by k and (2) they assume the *highly effective propagation*, i.e., the probability for good information spreading is either one or zero, whereas our decontamination model is more general since it allows arbitrary spreading probabilities. Moreover, we provide a far richer framework for studying the problem of containing viral spread on OSNs, where we consider *not only whether the initial set of nodes contaminated by misinformation is known to the defender, but also take into account the time allowed for the defender to contain the misinformation*. We thus believe the results from this work offer more insights into how to contain viral spread on OSNs under diverse constraints in practice.

In a nutshell, our main contributions are summarized as follows. First, we analyze an algorithm for the β-Node Protector problem, which greedily chooses nodes with the best marginal influence added to the current solution, and show that this algorithm selects only a small fraction of extra nodes from the optimal solution (Theorem 1). This result, indeed, provides us a better knowledge on the lower bound of the optimal solution in comparison with the $\left(1+\ln\frac{\beta N}{\epsilon}\right)$ factor suggested in [8]. Second, we show that the β_T^I-Node Protector problem is hard to approximate by a logarithmic factor, i.e., there is no polynomial-time algorithm that is guaranteed to find a solution of at most a logarithmic factor off the optimum solution. In normal graphs, we apply the greedy algorithm to the network restricted to T-hop neighbors of the initial set I and achieve a slightly better bound for the β_T^I-Node Protector problem. Third, we propose a community-based algorithm which returns a good selection of nodes to decontaminate the nodes in the network in an efficient manner. Finally, we conduct experiments on real-world traces including NetHEPT and Facebook networks [9], and empirical results show that both the greedy and community-based algorithms obtain the best results in comparison with other available methods.

2. Related work

The information and influence propagation problem on social networks was first studied by Domingos and Richardson in [10]. In this work, they designed viral marketing strategies and analyze the diffusion processes using a data mining approach. Later, Kempe et al. [11] formulated the influence maximization problem on a social network as an optimization problem. In their seminal work, they focused on the linear threshold and independent cascade models and proposed a generalized framework for both of them, as well as proving the problem of influence maximization with a k-node budget admits a $(1-1/e)$ approximation algorithm. Leskovec et al. [12] studied the influence propagation under the detection of outage break-out situation. In particular, they aimed to find the set of nodes in networks to detect the out-break, e.g., the spread of virus, as soon as possible. Chen et al. [13] improve the efficiency of the greedy algorithm and propose a new *degree discount* heuristic that is much faster and scalable.

Some attempts have been made in the light of containing the spread of misinformation. For instance, the concept of using benign computer worms to fight against another species has been studied in [14,15]. Most of these works focus on analyzing the performance of active worm containment in the traditional arena of worm propagation, where infectious computers use scanning strategies to find new victims in the IPv4 address space. Dubey et al. [16] conducted a study under the form of a network game focusing on quasi-linear model with various cost and benefit for competing firms. Bharathi et al. [17] modified the independent cascade model to better capture the competing campaigns in the network. Kostka et al. [18], from a game theory point of view, show that the first propagation spreading is not always advantageous. Recently, Budak et al. [7] considered the strategy of using "good" information dissemination campaign to fight against misinformation propagation in social networks. They formulated this as an optimization problem, proved that it is NP-hard, and then provided approximation guarantees for a greedy solution based on the submodularity property. This problem, indeed, can be considered as one variant of the family of problems we address in this work. In [8], Goyal et al. study information dissemination on social network. They provide an greedy algorithm and give a proof of factor $(1 + \ln \frac{\beta N}{\epsilon})$. However, that algorithm is a bicriteria one with an addictive error ϵ on the number of nodes, whereas our analysis suggests a deterministic bound which does not depend on any error parameter.

From the security perspective of OSNs, the information and influence propagation plays an important role in analyzing and designing strategies to counter misinformation such as rumor and computer malware. In [19], for example, Yan et al. conducted a comprehensive study of malware containment strategies, including both user-oriented and network oriented ones, on a moderate-sized online social network. Their work, albeit offering insights into the nature of malware propagation in realistic OSNs, was performed fully from an empirical perspective, and considered only a simple model for malware propagation. Tackling containment of viral spread of misinformation in OSNs, however, demands solutions with a stronger theoretic footing such that they are applicable to a variety of online social network structures and information dissemination models. Moreover, Xu et al. considered the problem of detecting worm propagation in online social networks, and showed that finding a minimal set of nodes to monitor for the purpose of traffic correlation analysis is an NP-complete problem [20]. Our work differs from theirs as we focus on containment, rather than detection, of misinformation propagation in OSNs.

3. Diffusion models and problem definition

In this section, we first define two models of influence propagation in online social networks, as well as a mechanism modeling how good information is disseminated in the network in order to contain misinformation. Under these models, we further formulate the β_T^I-Node Protector problem, which aims to find the smallest set of highly influential nodes in the decontamination campaign.

3.1. Propagation models

We first describe two types of information diffusion, namely the Linear Threshold and Independent Cascade models. These two practical propagation models have received great attention since their introduction in [11], and in this subsection, they are discussed with the same notations. For the sake of consistency, we call a node *active* if it is influenced by the misinformation either initially or sequentially from one of its neighbors, and *inactive* otherwise.

3.1.1. Linear Threshold (LT) model

In this model, the chance for a node v to adopt the misinformation from a neighbor w is determined based on the weight $b_{v,w}$ that satisfies $\sum_{w \in N(v)} b_{v,w} \leqslant 1$ for all w in the neighborhood $N(v)$ of v. Initially, each node $v \in V$ independently selects a threshold $\theta_v \in [0,1]$ uniformly at random. The goal of this threshold is to represent the weighted fraction of v's neighbors that must adopt the misinformation active in order for v to become active. Now, given the chosen thresholds for all nodes v's in V, the propagation progresses from an initial set of infected nodes I as follows: in step t, all active nodes in step $t-1$ remain active, and any inactive node v for which the total weight of its active neighbors is at least $\theta_v : \sum_{w \in N_{active}(v)} b_{v,w} \geqslant \theta_v$ is activated.

3.1.2. Independent Cascade (IC) model

In this model, any node v that became active in step t will have only one chance to activate each of its currently inactive neighbors w's, and the activation from v to a neighbor node w succeeds with a probability $p_{v,w}$. If node w has multiple newly activated neighbors, they will try to activate w sequentially in an arbitrary order. If any of these attempts succeeds, w is activated at time step $t+1$ and the same procedure continues further on w, i.e., w will try to activate its inactive neighbors. Again, any node is given only a single chance to influence its friends, thus if it fails to do so in time t, it is not allowed to activate its friends again in time $t+1$. Note that in both IC and LT models, the information can be propagated to multiple-hop neighbors.

3.1.3. Decontamination mechanism

The decontamination mechanism in our problem is coincident with the misinformation spreading model. In particular, once the good information spreads out from a particular set of nodes A_I, each node u in A_I will try to spread out the good information to its neighbor node v with the same influence probability $p_{u,v}$ (as in the underlying IC model), or with the same influence threshold θ_v (as in the LT model). We also assume that once a node is decontaminated with good information, it will no longer be influenced by the misinformation. Moreover, if good information and misinformation reach a node at the same time, the good information overrules the bad. This assumption makes sense for OSNs in reality as a good message could be announced to fix a specific misinformation.

3.2. Problem definition

We consider the following problem in this paper:

Definition 1 (β_T^I-*Node Protector*). Consider a social network represented by a directed graph $G = (V, E)$ and an underlying diffusion model (either LT or IC model). In the presence of misinformation spreading on G from an either known or unknown initial node set I, our goal is to choose the set $S \subseteq V$ of least nodes to decontaminate with good information so that the expected **decontamination ratio**, which is the fraction of uncontaminated nodes in the whole network, after T time windows, is at least β. Here $T \in \mathbf{N}$ and $\beta \in [0 \cdots 1]$ are input parameters.

Based on the settings of the initial set I and the time window T, we have the following four different variants of the Node Protector (NP) problems whose NP-hardness properties can be certified but are omitted here due to space limit.

1. β-NP: I is unknown and T is unconstrained ($T = \infty$).
2. β^I-NP: I is known and T is unconstrained ($T = \infty$).
3. β_T^I-NP: I is known and T is constrained ($T < \infty$).
4. β_T-NP: I is unknown and T is constrained ($T < \infty$).

3.3. Notations

Let $N = |V|$ be the total number of nodes in G. For any node $v \in V$ and any $A \subseteq V$, let $\sigma(v)$ and $\sigma(A)$ be the expected number of nodes that will be influenced by v and A, respectively, if v and A adopt the misinformation (or the dissemination of good information). Note that $\sigma(A)$ and $\sum_{v\in A}\sigma(v)$ are not necessarily the same in general, and $\sigma(v)$ can contain more nodes than just the neighbors of v since we are considering multiple-hop neighbors. For instance, consider a tree with $2N + 1$ nodes rooted as v with N level-1 nodes directly attached to v and N level-2 nodes, each of which attached to a level-1 node. Now, suppose v is influence by the misinformation, v will eventually influence its N level-1 neighbors. Each level-1 node can have half of the chance to influence its level-2 node, so $\sigma(v) = N + 1/2N = 3/2N > N = |N(v)|$.

In [11], Kempe et al. proved that σ () function is submodular under both IC and LT models, and thus the problem of selecting the set S of k nodes in the network that maximizes $\sigma(S)$ admits an $(1 - 1/e)$-approximation guarantee. Other works [7,21] also study this type of problem in some different settings and try to show the submodular property of the σ () function in these cases to obtain the same guarantee factor.

From a different viewpoint, our problem is complementary to the previous works as we are given a desired decontamination ratio β and the goal is to find the set S of least nodes under this constraint. We stress that while other studies try to prove the submodular property to get the well-known $(1 - 1/e)$ factor, we indeed use this factor to provide a new perspective into the problem. Specifically, we use this guarantee to derive a better lower-bound on the number of nodes in the optimal solution, as described in the next section.

Algorithm 1. GVS algorithm for the β-Node Protector problem

```
Input: Network $G = (V, E)$, threshold $\beta \in (0, 1]$;
Output: A set $S \subseteq V$ satisfies $\sigma(S) \geqslant \beta|V|$;
$k \leftarrow 1$;
$S_k \leftarrow \emptyset$;
while ($\sigma(S_k) < \beta|V|$) do
    $v \leftarrow \arg\max_{u\in V\setminus S_k}\{\sigma(S_k \cup \{u\}) - \sigma(S_k)\}$;
    $k \leftarrow k + 1$;
    $S_k \leftarrow S_k \cup \{v\}$;
end while
Return $S$.
```

4. A bounded method for β-Node Protector

When the initial infected set I is unknown and the time window T is unconstrained, the β_∞ (or simply β)-Node Protector problem asks for the smallest set of nodes from which disseminating good information helps achieve a decontamination ratio of at least β *at the end of the process*, when no more nodes can be influenced or decontaminated. This is the case that usually occurs in practice, especially for large OSNs, and is also the most difficult case to solve. The main source of difficulty here is that the lack of knowledge about the initial set I does not enable us to wisely choose nodes to decontaminate, and thus, we have to do it blindly with the hope that we could have a good solution. Moreover, due to its NP-hardness nature, it seems unrealistic for one to expect an optimal algorithm for this problem.

4.1. A greedy algorithm with a bounded solution

We analyze GVS (*Greedy Viral Stopper*), a greedy solution to the β-Node Protector problem, which utilizes a modification of the well-known Hill-Climbing (HC) algorithm [22]. At each round of the algorithm, we add a new node v with the maximal marginal gain $\sigma(S + v) - \sigma(S)$ to the current set S until the fraction of safe nodes reaches β. By doing so, we can show that this solution is within a small amount of nodes in addition to the optimal solution (Theorem 1). Algorithm 1 describes the GVS algorithm.

Theorem 1. *Algorithm 1 returns a solution S of K nodes to the β-Node Protector problem that expectedly satisfies*

$$K \leqslant |OPT| + \max\left\{0, \frac{\beta N}{\delta e} - \frac{\Delta}{\delta} + 1\right\},$$

where $N = |V|$ is the total number of nodes in the network, $\Delta = \beta N - \sigma(S_{K-1}), \delta = \min_{i=1,\ldots,K-2}\{\sigma(S_{i+1}) - \sigma(S_i)\}$ and OPT is an optimal solution to the β-Node Protector problem.

Proof. Let us first describe the notations used in this proof. Let $S_i \quad (i = 1 \ldots K)$ be the set produced at the ith step of Algorithm 1 (note that $S \equiv S_K$ by this definition). For any integer k, let $Opt_\sigma(k)$ be the maximum value of $\sigma(A)$ over all sets $A \subseteq V$ of $|A| = k$ nodes, i.e., $Opt_\sigma(k) = \max_{A \subseteq V, |A|=k}\{\sigma(A)\}$. Furthermore, let $q = |Q|$, where Q is the optimal solution set with the lowest $\sigma(Q)$ that exceeds βN, i.e., $Q = \arg\min_{A \text{is an OPT}, \sigma(A) \geqslant \beta N}\{\sigma(A)\}$. It follows from the above definitions that

(i) $q = |OPT|$ for any optimal solution set OPT.
(ii) $Opt_\sigma()$ and σ () are nondecreasing functions, and $\forall A \subseteq V, \sigma(A) \leqslant Opt_\sigma(|A|)$.
(iii) There lie no $Opt_\sigma(k)$'s ($\forall k = 1 \ldots K$) strictly between βN and $\sigma(Q)$ (otherwise it will violate the definition of Q).

We are now ready to prove the theorem. Since Algorithm 1 terminates at the Kth step, it follows that $\sigma(S_{K-1}) < \beta N$. We consider the following cases:

Case 1: $Opt_\sigma(K-1) < \beta N$. When $Opt_\sigma(K-1) < \beta N$, it implies that any set with even $K-1$ nodes is not sufficient to achieve the desired disinfection goal. Therefore, the optimal solution Q must contain $q > K-1$ nodes. Moreover, $q \leqslant |S| = K$ since S is a regular solution. This means $|Q| = K$, which in turns implies that S is also an optimal solution.

Case 2: $\beta N \leqslant Opt_\sigma(K-1) \leqslant \sigma(Q)$. Due to (iii) and the definitions of Q and $Opt_\sigma(\cdot)$, this case can only be valid either when (a) $Opt_\sigma(K-1) = \beta N$ or (b) $Opt_\sigma(K-1) = \sigma(Q)$.

When (a) $Opt_\sigma(K-1) = \beta N$, it follows that the optimal solution Q must span exactly βN nodes since $\sigma(Q)$ is the closest to βN. Note that this does not necessarily indicates that $|OPT| = K-1$. Instead, what it implies is $\beta N = Opt_\sigma(K-1) = Opt_\sigma(K-2) = \ldots = Opt_\sigma(q) = \sigma(Q)$.

When (b) $Opt_\sigma(K-1) = \sigma(Q)$, it again infers that $\sigma(Q) = Opt_\sigma(q) = \ldots = Opt_\sigma(K-1)$ due to (ii). Now, if $q = K-1$, this greedy algorithm will incur just one more node than the optimal solution. Otherwise, the analysis of Case 3 can be applied in a very similar manner and consequently, we obtain the same result $K \leqslant q + \left(\frac{\beta N}{\delta e} - \frac{\Delta}{\delta} + 1\right)$ for both situations.

Case 3: $\sigma(Q) < Opt_\sigma(K-1) \leqslant N$. This is our main case to handle. What we know in this case is $|OPT| = q \leqslant K-1$, implying $Opt_\sigma(q) \leqslant Opt_\sigma(K-1)$. We need to bound the size difference between S_{K-1} and S_q, or in other words, bounding $K-1-q$. To do so, we will use the $(1-\frac{1}{e})$ approximation result in [22] (note that all K steps of Algorithm 1 follow the HC algorithm, and hence, this guarantee follows naturally), giving $(1-\frac{1}{e})Opt_\sigma(q) \leqslant \sigma(S_q) \leqslant \sigma(S_{K-1}) = \beta N - \Delta$. Therefore, $\sigma(S_{K-1}) - \sigma(S_q) \leqslant (\beta N - \Delta) - (1-\frac{1}{e})Opt_\sigma(q)$. In addition, since $Opt_\sigma(q) \geqslant \sigma(Q) \geqslant \beta N$, the above inequality becomes $\sigma(S_{K-1}) - \sigma(S_q) \leqslant (\beta N - \Delta) - (1-\frac{1}{e})\beta N = \frac{\beta N}{e} - \Delta$.

In order to lower bound $\sigma(S_{K-1}) - \sigma(S_q)$, we observe that every time a node v is added into the solution set S_i, the expected number of decontaminated nodes increases at least by $\min\{\sigma(S_{i+1}) - \sigma(S_i)\} \geqslant \delta$ for $i = q, \ldots, K-2$. Hence $(K-1-q)\delta \leqslant \sigma(S_{K-1}) - \sigma(S_q) \leqslant \frac{\beta N}{e} - \Delta$, which implies $|S| \equiv K \leqslant |OPT| + \left(\frac{\beta N}{\delta e} - \frac{\Delta}{\delta} + 1\right)$. This bound also concludes the proof. □

Theorem 1 implies the solution returned by GVS is within a linear factor of βN extra from the optimal solution, given the desired decontamination ratio β. Intuitively, the abitrary selection of any βN nodes in the network is always sufficient for our problem; however, this lower bound implies that GVS, indeed, selects at most $\frac{1}{e} \approx 36\%$ of this many nodes in addition to the optimal solution. Moreover, the bigger β, i.e., the smaller the number of misinformation nodes we allowed, the more nodes we have to protect, and vice versa. The bound in Theorem 1 nicely reflects this intuition: when β is bigger, the range for K in the right hand side (RHS) gets larger, which allows K to gets bigger as more nodes needed to be decontaminated. Vice versa, when β gets smaller, the range for K reduces as fewer nodes need to be protected.

4.2. Time complexity

It is easy to see that Algorithm 1 will terminate after at most βN step in the worst case and the most expensive task is to find $\max_{u \in V \setminus S}\sigma(u)$ in Step 4 of the algorithm. In [23], the authors presented an efficient way to handle this task in (approximatively) $O(NM_0\Delta)$ where M_0 is the number of samples and Δ is the maximum complexity for finding $\sigma(v)$ for any node v. Therefore, the total time complexity of this bounded algorithm is $O(N^2M_0\Delta)$.

5. Algorithms for the β^I- and β^I_T-Node Protectors problem

In this section, we study the family of β^I_T-Node Protector problems where the initial infected set I is known and the time step T is either constrained or unconstrained. Due on its NP-hardness nature, it seems unrealistic for one to find an optimal solution to the β^I_T-Node Protector problem in a tractable manner. We further show, by Theorem 2, that this problem in general is hard to approximate with a logarithmic factor via a similar reduction procedure from the Set Cover problem [24]. This result shows how difficult this problem is as it implies the nonexistence of any logarithmic approximation algorithm for the β^I_T-Node Protector problem, under the assumption that $P \neq NP$.

5.1. Inapproximability of β^I_T-Node Protector

The inapproximability of the β^I_T-Node Protector problem can be shown via a modified reduction from the Set Cover problem. Given an instance $\mathcal{I}_{SC} = \{\mathcal{U}, \mathcal{S}\}$ of the Set Cover problem, where $\mathcal{U} = \{u_1, u_2, \ldots, u_l\}$ is the set of l elements and $\mathcal{S} = \{S_1, S_2, \ldots, S_t\}$ is a collection of t subsets of $\mathcal{U}$, we (1) place $S_1, \ S_2, \ \ldots, \ S_t$ on the left hand side (LHS) and $u_1, \ u_2, \ \ldots, \ u_l$ on the RHS, and (2) connect a directed edge from S_i to u_j with probability 1 if $u_j \in S_i$. Note that the direction simply means the infection only propagates from one side to another, but not vice versa. We

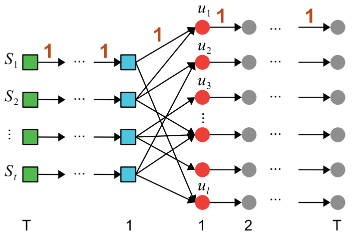


Fig. 1. A modified reduction from the Set Cover problem. All edges have probability 1.

additionally create $T-1$ copies of $S_1,\ S_2,\ \ldots,\ S_t$ as well as $T-1$ copies of $u_1,\ u_2,\ \ldots,\ u_l$, and then tie these newly created nodes accordingly to S_i's and u_j's to make t directed chains on the LHS and l corresponding directed chains on the RHS, with probability 1 on each edge (Fig. 1). Now, the initially infected set I is set to be $\{u_i|i=1,\ldots,l\}$ (red nodes) and we want all nodes in the RHS to be inactive at the end of the process, i.e., $\beta=1$. Let us, for short, name this instance β_{SC} and denote $n_{sc}=t+l$. Note that $N=n_{sc}T$ in β_{SC}. Using this reduction, one can show the following result:

Theorem 2. *The β_T^I-Node Protector problem cannot be approximated in polynomial time to a factor of $c\ln N$, where c is some constant and N is the number of vertices in the graph, under the assumption $P\neq NP$.*

Proof. Before proving this theorem, it is easy to see that the optimal solution of a Set Cover instance and its corresponding β_{SC} have the same size, i.e., $|OPT(SC)|=|OPT(\beta_{SC})|$. Moreover, there exists an optimal solution for β_{SC} not containing any u_i or its copies (since we can always exclude u_i and include the node representing any of the set S_j that u_i belongs to, which leads to a better or equivalently good solution). In other words, there exists an optimal solution containing only subsets from $\mathcal{S}$. Now, if there is a polynomial time algorithm $\mathcal{A}$ that can approximate β_{SC} to $c\ln N$ for some constant c, i.e., it can find a solution S for β_{SC} satisfying $|S|\leqslant c\ln N\times|OPT(\beta_{SC})|$. Since $N=n_{sc}T=(t+l)T$, we can rewrite this inequality as

$$\begin{aligned}|S|&\leqslant c\ln N\times|OPT(\beta_{SC})|=c\ln(n_{sc}T)\times|OPT(\beta_{SC})|\\&\leqslant(d+c)\ln n_{sc}\times|OPT(SC)|=c'\ln n_{sc}\times|OPT(SC)|\end{aligned}$$

where d is any constant satisfying $d\geqslant\frac{c\ln T}{\ln n_{sc}}$, given the quantities T,t and l of the Set Cover and β_{SC} instances. This means algorithm $\mathcal{A}$ can approximate an instance of Set Cover problem to a factor of $c'\ln n_{sc}$ in polynomial time, where c' is some constant. This implication does not appear to hold true under the assumption P $\neq$ NP [25]. Thus, the conclusion follows. □

5.2. Algorithms for β^I and β_T^I-Node Protector

We next present solutions for β^I- and β_T^I-Node Protectors on general networks. The spirit of our approaches for these cases is also based on HC algorithm, however, the searching space is significantly reduced due to the knowledge of the initial set I. In particular, we apply GVS algorithm on the network restricted to T-hop neighbors of the initial set I. This provides a slightly better term extra from the optimal solution in comparison with the case of β-Node Protector. Our approach is based on the following crucial observation: once the infected set is known, the total set of nodes possibly influenced by I is reduced to $N_T(I)$ while the nodes in $V\setminus N_T(I)$ will never be active, where $N_T(I)=\bigcup_{u\in I}N_T(u)$ and

$$N_T(u)=\begin{cases}\{v\in V|u\text{ can reach }v\text{ within }T\text{ hops}\}, & T<\infty\\ \{v\in V|u\text{ can reach }v\}, & T=\infty\end{cases}$$

Hence, once given an initial infected set I and the desired disinfection ratio β in T time windows, the algorithm first identifies $N_T(I)$ and then executes GVS (Algorithm 1) on the induced graph $G[N_T(I)]$ with the new disinfection ratio $\beta'=\beta-\frac{N-|N_T(I)|}{N}$, since these $N-|N_T(I)|$ nodes are out of reach of I. If $\beta\prime\leqslant 0$, it means that the fraction of nodes outside of $N_T(I)$ is itself sufficient and thus, we do not have to execute the algorithm. Therefore, we focus on the case $\beta'\geqslant 0$. By using the similar analysis as in the case of β-Node Protector, we can derive the following result

Theorem 3. *Algorithm 1 on the induced graph $G_{N_T(I)}$ returns a solution S of K nodes for β_T^I-Node Protector that expectedly satisfies*

$$K\leqslant|OPT|+\max\left\{0,\frac{\beta'|N_T(I)|}{\delta e}-\frac{\Delta'}{\delta}+1\right\},$$

where $\delta=\min_{i=1,\ldots,K-2}\{\sigma(S_{i+1})-\sigma(S_i)\},\Delta'=\beta'N-\sigma(S_{K-1})$, and OPT is an optimal solution set for β_T^I-Node Protector problem.

Proof. The proof is similar to that of β-Node Protector and is omitted here. □

6. Experimental results

In this section, we show the experimental results of the GVS algorithm on three real networks including the NetHEPT, NetHEPT_WC and the Facebook social networks. We want to demonstrate the followings (1) how the GVS algorithm works on β- and β_T^I-Node Protector problems in comparison to other available methods and (2) the expected lower bounds of the optimal solutions between ours and those suggested by the $(1+\ln\frac{\beta N}{\epsilon})$ factor [8].

We also planned to compare our results against those of [7]. However, since the dissemination probabilities in our models are distributed in the range $[0,1]$, whereas [7] assumes a highly effective decontamination campaign (i.e., good information spreads out with an absolute probability, i.e., $p_{u,v}=1$ if there is an edge from u to v, and zero otherwise), it does not seem appropriate to do so. In what follows we assume the IC information propagation model.

6.1. Datasets

6.1.1. NetHEPT and NetHEPT_WC

The NetHEPT network is a widely used dataset for testing information diffusion [21,26]. This dataset contains information, mostly the academic collaboration from the "High Energy Physics – Theory" section on arXiv where nodes stand for authors and links represent the coauthorship. In their deliverable, the NetHEPT network contains 15,233 nodes and 31,398 links, and the probabilities on edges are assigned either uniformly at random (for NetHEPT) or with a *weighted cascade* model (for NetHEPT_WC), where $p(u,v) = 1/d_{in}(v)$ with $d_{in}(v)$ being the in degree of node v. Note that the weighted cascade model is a special case of the IC model when the probability for each edge is predetermined.

6.1.2. Facebook network

This dataset contains the friendship information among the Facebook user accounts in the New Orleans region, spanning from September 2006 to January 2009 [9]. To collect the information, the authors created several Facebook accounts, each of which joined the regional network and crawled the network in a breath-first-search fashion. The data set contains more than 63 K nodes (users) connected by more than 1.5 million friendship links with an average node degree of 23.5. In our experiments, the propagation probability for each link connecting to users u and v is proportional to the communication frequency between u and v, normalized over the entire network.

6.1.3. Setup

We compare the GVS algorithm against the following alternative solutions: (1) the *Random* method, which include nodes added randomly to the current solution until the stopping criterion is met; (2) the *High degree* method, which adds nodes with the highest weighted degree to the solution until the stopping criterion is met; (3) the *DiscountIC* method, which is based on the weighted discount for the IC model suggested in [21], and (4) the *Page Rank* method, which adds nodes sequentially to the solution based on their PageRank until the stopping criterion is satisfied.

In all experiments, the Monte Carlo simulation for estimating expected influence is averaged over 1000 runs for consistency. Since the execution of the GVS method is expensive as shown in the running time analysis, we just conduct test cases on small values of β, ranging from 0.01 to 0.37. At any value of β, we run all methods independently and report the number of selected nodes suggested by each of them.

6.2. Number of selected nodes

6.2.1. Results on the β-Node Protector problem

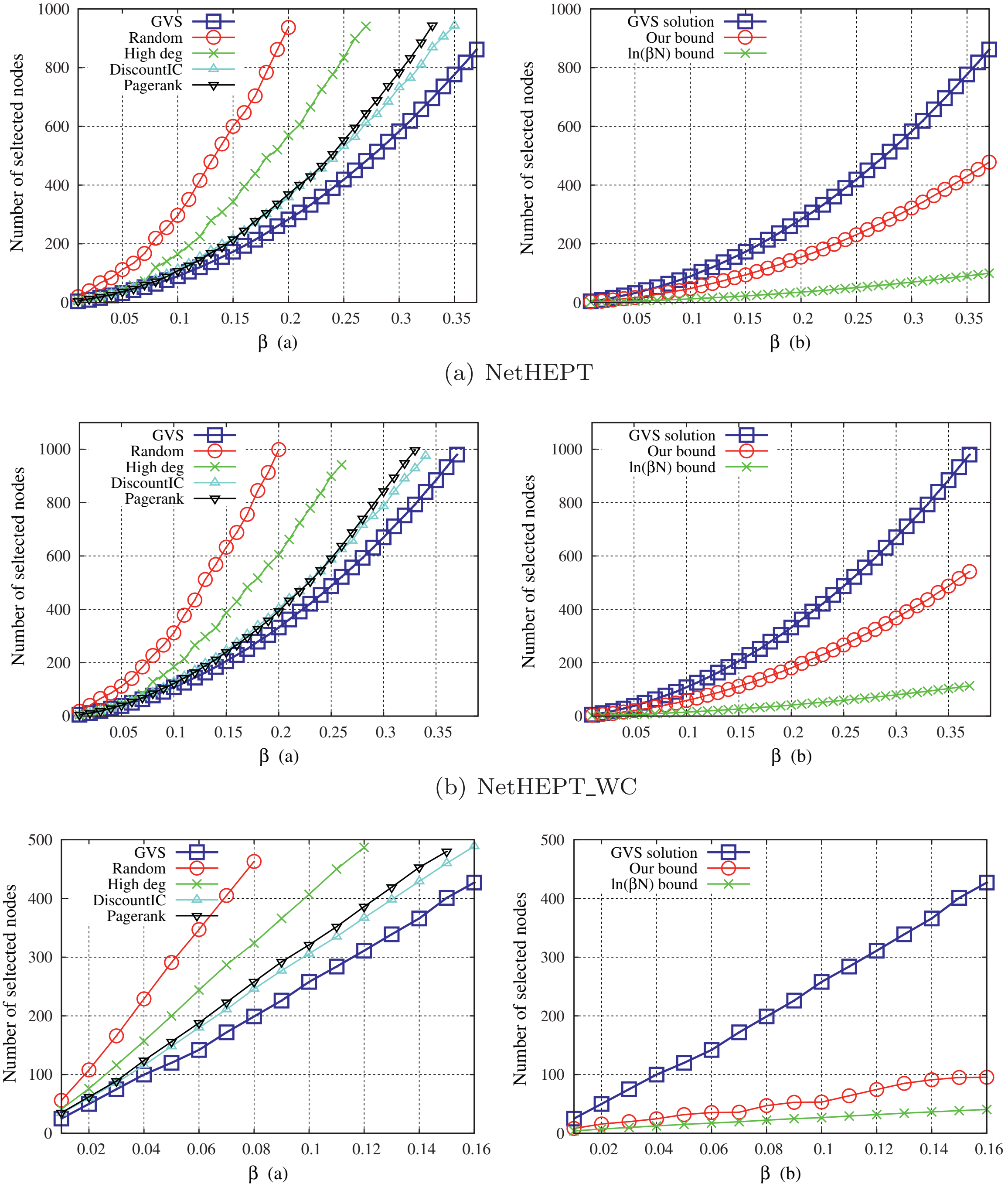
Recall that the ultimate goal of our problem is to choose the set of least nodes so that a dissemination ratio of at least β over the entire network is achieved. The left charts of Figs. 2a–c report the performances of the five methods on the β-Node Protector problem for the three different datasets. As depicted in those figures, the number of selected nodes returned by the GVS algorithm is always the smallest among all the methods considered. In particular, the solution returned by GVS is roughly 24% better than that from the second best method, *DiscountIC*, is 75% better than that from *Highest Degree*, and is more than 1.5 timers better than the solution returned by *Random*.

We observe that the behaviors of all five methods are nearly the same on the NetHEPT and NetHEPT_WC datasets although the number of selected nodes on the NetHEPT_WC dataset is slightly smaller. Moreover, the number of selected nodes tends to curve up as the target dissemination ratio β gets larger. On the Facebook dataset, GVS, again, outperforms the other methods in term of the size of the selected node set and is much better than the *Random* and *Highest Degree* methods. While the *Random* method, not surprisingly, performs the worst in the pool, we had expected the *Highest Degree* method to have a better performance. Its poor performance can be explained by the sparseness of the real online social network: although the nodes of the highest degrees could influence more nodes directly, they are not necessarily the most influential ones possibly because of the low chances they can influence each of their neighbors. In fact, this is the case since edges with high probabilities mostly connect low degree nodes in the Facebook network. In addition, unlike what we have observed for the other datasets, the number of nodes returned by each of the methods on the Facebook network increases only linearly as β becomes bigger.

6.2.2. Results on the β^I and β^I_T-Node Protector problems

We next look at the performances of GVS and the other methods when working on the β^I and β^I_T-Node Protector problems. In particular, we randomly choose 15% of all nodes to be I, the initial source of misinformation propagation, and vary T, the time window, between 5 and ∞. We restrict the scope of the GVS algorithm on the reduced network $G[N_T(I)]$, and provide I as part of the input for the other methods as well. The numbers of nodes and edges contained in these restricted networks $G[N_T(I)]$'s for the β^I_T-Node Protector problem are 6645 and 18,236 for NetHEPT, 8647 and 21,091 for NetHEPT_WC, and 21,816 and 865,930 for the Facebook dataset, all respectively. For the β^I-Node Protector problem setting, these numbers are 7315 and 22,189 for NetHEPT, 9745 and 26,352 for NetHEPT_WC, and 26,816 and 1 M for the Facebook dataset, all respectively.

The results are reported in the left charts of Fig. 3a–c. As expected, the numbers of selected nodes returned by the GVS algorithm on the reduced networks outperform the other alternative solutions with bigger gaps among them. On average, the solution returned by GVS is 64% better than that by *DiscountIC*, 1.2 times better than that from the *Highest Degree* method, and up to 2.3 times better than the solution found by the *Random* method. These results are also consistent with what have been observed in the previous test case. We also notice the reduction on the numbers of nodes to be selected by all five methods, particularly GVS. The number of nodes chosen by the GVS algorithm is reduced by at least one half for each of the three cases. Of course, this is what one should expect once the knowledge of I is provided, and especially when the sizes of the restricted networks are much smaller. The empirical result charts for the β^I-Node Protector problem are highly similar to what we have seen in Fig. 3 and are thus excluded here.



(a) NetHEPT

(b) NetHEPT_WC

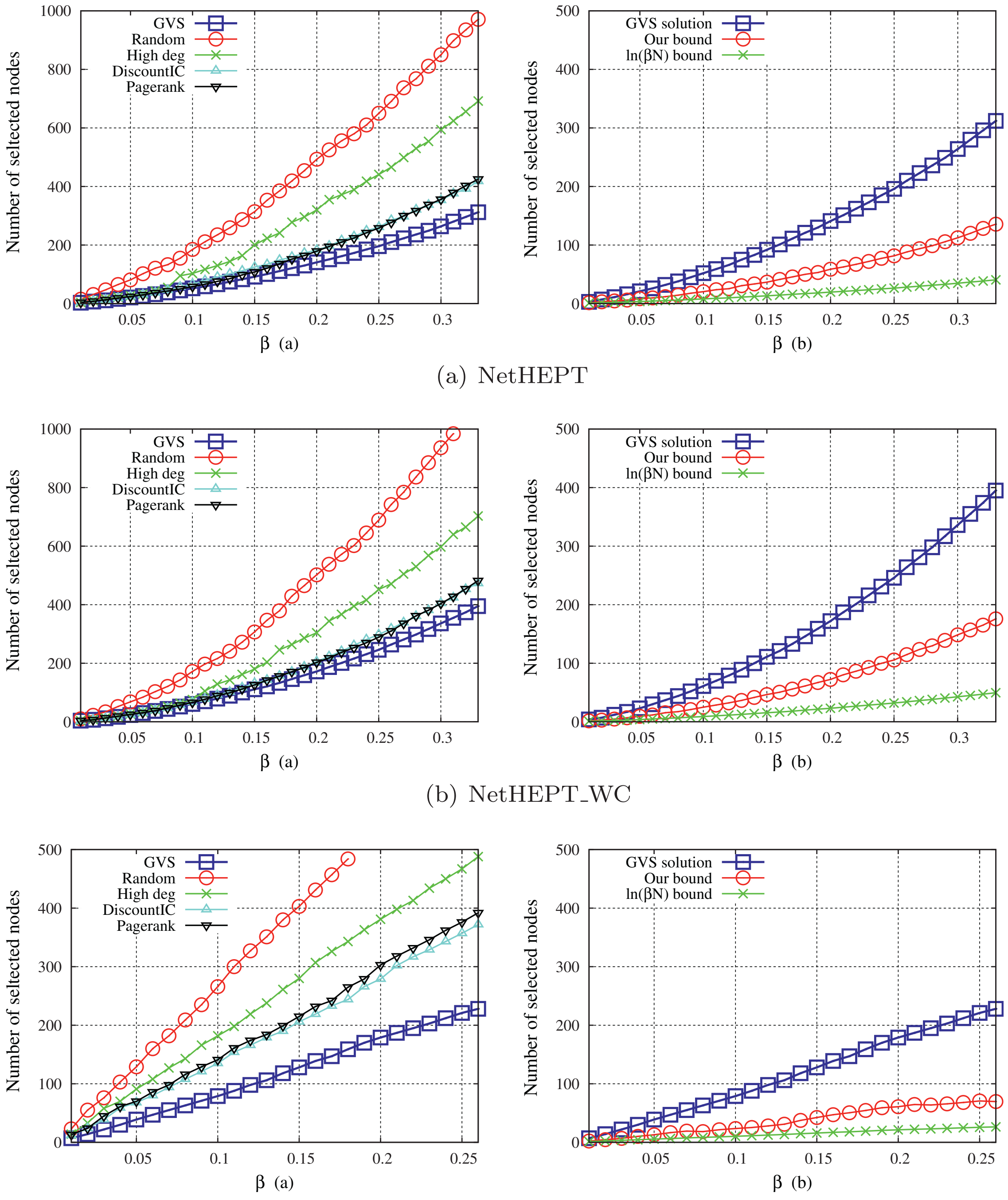
(c) Facebook

Fig. 2. Results of GVS algorithm for β-Node Protector on real social networks (left figures) and the expected lower bounds of the optimal solution (right figures).

There is room for improvement here on how to wisely choose nodes when I is the set of some specific targets such as high-degree nodes, or nodes with low or average degrees. While they do sound interesting, we believe that is not what we are aiming at here, and thus, it would be treated in our future work.

6.3. Lower bounds of the optimal solutions

We further investigate how big the size of the optimal solution set would expectedly be once we are provided with the greedy solution S from the GVS algorithm. Recall that our result is $|S| \equiv K \leqslant |OPT| + \max\left\{0, \left(\frac{\beta N}{\delta e} - \frac{\Delta}{\delta} + 1\right)\right\}$.



(a) NetHEPT

(b) NetHEPT_WC

(c) Facebook

Fig. 3. Results of GVS algorithm for β_5^I-Node Protector on real social networks (left figures) and the expected lower bounds of the optimal solution (right figures).

Goyal et al. [8] states that $K \leqslant \left(1+\ln\frac{\beta N}{\epsilon}\right)|OPT|$, where $\epsilon \geqslant 1$ is the additive error to the number of nodes to be decontaminated. Given a solution S by the GVS algorithm, both this result and ours provide the knowledge on the lower-bound of the optimal solution OPT. Theoretically, neither of them dominates the other on all network instances (for example, a clique with a probability of $(1-\eta)$ on all edges is more suitable for the $\left(1+\ln\frac{\beta N}{\epsilon}\right)$-factor while real social networks are more suitable for ours, as we shall see below). Of course, a larger and tighter value for this lower bound is more desirable as it tells us how large the size of the optimal solution should be. Thus, we also want to know how the two results look like in real-world networks. Here, $\epsilon = 1$ since in term of nodes, this error should be an integer greater or equal to one.

As revealed in the right charts of Figs. 2 and 3, our lower bounds (indicated in red circles) are usually larger than that of [8] in all test cases, and consequently provide a more meaningful insight into the expected size of the optimal solution: for any given β, the size of the optimal solution should lie somewhere in between the region induced by the GVS solutions and our lower bounds. This also provides a new point of view into the problem as it ensures that the optimal solution is not too far away from the one returned by the greedy algorithm. Note that our result does not necessarily imply the existence of any constant or logarithmic factor approximation algorithms.

7. A community-based heuristic algorithm

As described in the experiments, the GVS algorithm provides good solutions for both the β- and β_T^I-Node Protector problems in comparison with the other methods. One of its down sides, however, is the extremely slow execution due to the expensive task of estimating the marginal influence when a node is added to the current solution. Even with available speed-up techniques provided in [21,26,23] for estimating this marginal gain, GVS still takes a long time to finish its tasks, especially on the Facebook social network (e.g., more than 5 h with a commodity PC). This means GVS, despite its good performance, might not be the best method for analyzing large-scale online social networks, particularly when the execution time is also a constraint. This drives the need for a more desirable approach which can return a good solution in a timely manner.

To this end, we take into account a notable phenomenon that is commonly exhibited in large-scale online social networks: the property of containing community structures, i.e., these networks naturally consist of groups of vertices with denser connections inside each group and fewer connections crossing among groups, where vertices and connections represent network users and their social interactions, respectively. Roughly speaking, a community on social networks usually consists of people sharing common interests who tend to interact more frequently with other members in the same community than with the outside world. The knowledge of the network community structure, as a result, provides us a much better understanding about its topology as well as its organization principles. Community detection methods and algorithms can be found in a comprehensive survey by Fortunato et al. [27].

Social-based and community-based algorithms utilizing network communities have been shown to be effective, especially when applied to online social networks [28–30]. Taking into account this great advantage of community structures, we propose a community-based heuristic method that can return a reasonable solution in a timely manner. More specifically, our method consists of two main phases: (1) the *community detection* phase, which quickly reveals the underneath network community structure and (2) the *high-influence node selection* phase, which effectively selects nodes of high influence. The detailed algorithm is described in Algorithm 2.

Algorithm 2. A community-based algorithm for the β-Node Protector problem

```
Input: Network $G = (V, E)$, threshold $\beta \in (0, 1]$;
Output: A set $S \subseteq V$ satisfies $\sigma(S) \geqslant \beta|V|$;
Use Blondel's algorithm [31] to find community structure in $G$;
Let $\mathcal{C} = \{C_1, C_2, \ldots, C_p\}$ be the disjoint network communities with $|C_1| \geqslant |C_2| \geqslant \cdots \geqslant |C_p|$;
$S \leftarrow \emptyset$;
for $i$ from 1 to $p$ do
    $S_i \leftarrow \emptyset$;
    while ($\sigma(S_i) < \beta|C_i|$) do
        $v \leftarrow \arg\max_{u \in V \setminus S_i}\{\sigma(S_i + u) - \sigma(S_i)\}$;
        $S_i \leftarrow S_i \cup \{v\}$;
    end while
    $S \leftarrow S \cup S_i$;
    if $\sigma(S) \geqslant \beta|V|$ then
        break;
    end if
end for
Return $S$.
```

7.1. Community detection

As described in Algorithm 2, detecting the network communities is the first phase and also is an important part of our method. A precise community structure that naturally reflects the network topology will help the selection of influential nodes in each community to be more effective. Because the detection of network communities is not our focus in this paper, we utilize an community detection method proposed by Blondel et al. [31] whose performance has been verified thoroughly in the literature [32].

There are some good features of this detection algorithm that nicely suit our purposes. First, it returns a community structure whose links within a single community usually have high probabilities, and those that come across communities are often of low probabilities. This is to say, the information propagation within each community occurs with a high probability whereas the chance for misinformation to spread between communities is relatively small. Second, the size of each community is much smaller in comparison with the whole network, and nodes in each community are usually (weakly) connected to each other. Moreover, they often have small diameters, i.e., nodes in them can reach each other within only a few hops. Finally, it execution is reasonably fast, which means it would not add too much time to the entire process.

7.2. High-influence node selection

The first phase provides us a community structure $\mathcal{C}$ as a partition of V into disjoint subsets $C_1, C_2, \ldots, C_p$, and we need to select nodes from these subsets so that a decontamination ratio of β is achieved. For simplicity, we assume that the communities are sorted in a non-increasing order of their cardinalities.

Since edges crossing between communities usually have low probabilities, it follows that a node from a community often has only a low chance to spread out misinformation (or good information if it is decontaminated) to another node in a different community. Therefore, our problem can be regarded as the selection of nodes in each community to decontaminate so that a decontamination ratio of β is achieved within each community, and hence achieving a decontamination ratio of β over the entire network. Intuitively, one would think of applying the GVS algorithm to each community to find the set of influential nodes to decontaminate. In fact, that is the approach we adopt here. For each community C_i, we greedily select nodes providing the maximal marginal gain and add it to S_i as well as the final solution S, until the stopping criterion is met. The motivation behind our approach is due to the stronger influence within each community and the much smaller size of each community in comparison with the whole network. Therefore, the selection process of high-influence nodes should not be as computationally prohibitive as it used to be.

7.3. Experimental results

In this subsection, we report the experimental results of the heuristic algorithm alongside those from the aforementioned methods. We demonstrate the followings: (1) the number of selected nodes and (2) the execution time. The experimental setup is kept the same as in Section 6. The numbers of communities detected by Blondel's algorithm on NetHEPT, NetHEPT_WC and Facebook are 1841, 1839 and 260, respectively. We remove the results from the *Random* method from the charts due to its poor performance and to make the plots more visible.

Simulation results are reported in Fig. 4. As depicted in these subfigures, the numbers of nodes selected by the community-based method (Community – in red circles) are highly competitive in comparison with those of the other methods, let alone that its performance tends to get much better as more nodes need to be decontaminated. In particular, the community-based approach is slightly outperformed by the others for small values of $\beta \in [0 \cdots 0.18]$ on both NetHEPT and NetHEPT_WC datasets and $\beta \in [0 \cdots 0.9]$ on the Facebook network. However, it performs much better than the other methods as the targeted dissemination ratio β gets larger. On average, the solution provided by the community-based method is roughly 16%, 41% and 22% better than those by the GVS, *High Degree* and *DiscountIC* methods for $\beta \in [0.2 \cdots 0.35]$ on NetHEPT and NetHEPT_WC, respectively, and is nearly 10% better than the greedy method on the Facebook dataset for $\beta \in [0.09 \cdots 0.16]$. Moreover, this quantity tends to increase only linearly in a long run on all of the test networks, unlike the expensive superlinear shapes of the others.

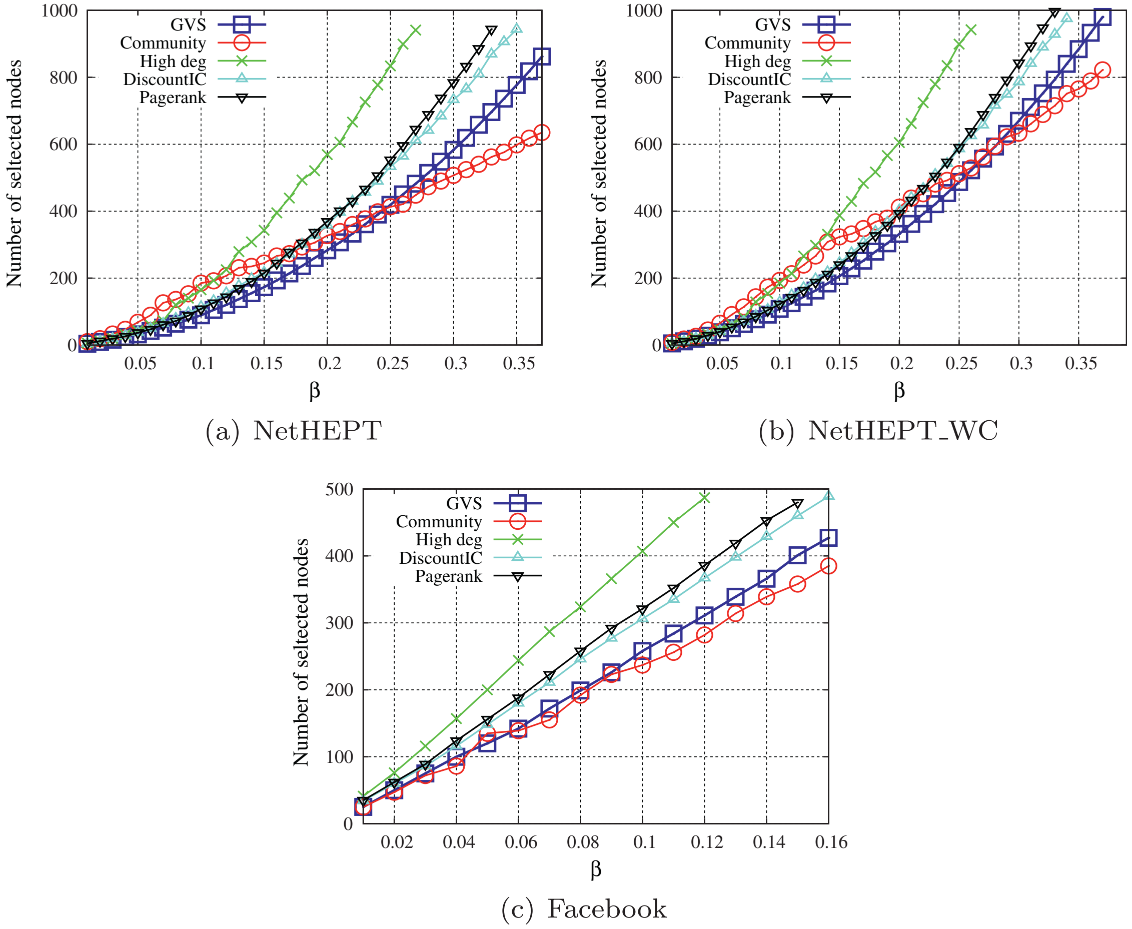
There are reasons behind both the advantages and disadvantages of the community-based method observed in our experiments. A careful look into the community structures of the three networks reveals that both NetHEPT, NetHEPT_WC and Facebook, in fact, feature only a few large-sized communities while containing a lot of small-sized ones, most of which are of cardinalities 6 and 8 (note that this is not a surprising observation as it intuitively agrees well with the findings of [27,33]). Moreover, the most influential nodes are usually scattered among different communities. Therefore, when the target decontamination ratio β is small, the GVS algorithm can quickly find these influential nodes while the community-based algorithm has to obey the rule of selecting influential nodes in larger communities. This explains the disadvantages of the community-based method in the smaller range of β. On the other hand, however, when more and more nodes need to be decontaminated to achieve the target decontamination ratio, the community-based algorithm can simply pick the most influential nodes from communities of smaller sizes (where they can easily influence the whole community), while the GVS algorithm may have to select more nodes from the other places to satisfy the criterion (see Table 1).

We next discuss the effect of network communities in our problems. The general belief holds that while the local "stars" are important for vaccination, decontamination should also target those nodes with edges across communities, since (1) they are typically only a small population and (2) decontaminating them will keep misinformation within small circles. This, however, is not what has been observed in our experiments. The reason can be explained as follow: as the community structure discovered has high internal propagation probabilities within each community and low propagation probabilities across different communities, the bridge nodes, generally speaking, have only low chances of spreading the misinformation to their neighboring communities; by contrast, the local "stars" possess much higher influence in spreading the information to the entire community. This is, perhaps, our most interesting finding that is somewhat in contradiction with the general belief.

7.4. Running time

We next compare the execution times of the community-based algorithm (the time spent on community detection is also included) and the other methods. In compensation for its good performance, GVS consumes a much longer time than the other methods on each of the datasets, especially on the Facebook network for which it takes more than 5 h to find the solution. While the other methods take fair amounts of time (from 1 to 21 min) for analyzing these average-sized datasets, they pose potential possibilities to consume much more time on larger social networks. In contrast, the community-based method, by leveraging the network community structure, is able to reduce the processing time significantly, while maintaining a competitive performance. Unlike GVS, however, this algorithm does not provide any guarantee on the quality of the solution relative to the optimal one.

In conclusion, we believe that GVS is one of the best informative methods for finding highly influential nodes



(a) NetHEPT

(b) NetHEPT_WC

(c) Facebook

Fig. 4. Results of the community based method on social networks.

Table 1
Running time of five methods.

	NetHEPT	NetHEPT_WC	Facebook
GVS	33.1 min	36.1 min	5 h
Community	**10 s**	**11 s**	**2.4 min**
High Deg	1.18 min	1.2 min	4.3 min
DiscountIC	4.3 min	4.3 min	20.3 min
PageRank	14.4 min	14.4 min	21 min

in small social networks when the running time is not a concern, whereas the community-based method can be regarded as a good heuristic-based method for detecting those important nodes for large-scale social networks.

8. Conclusion and discussion

We study the family of β_T^I-Node Protector problems which aim to find the set of least nodes from which disseminating "good" information provides a decontamination ratio of at least β over the entire network. We analyze an algorithm for the β-Node Protector problem that greedily adds nodes with the highest influence gains to the current solution, and show that this algorithm selects only a small fraction of nodes in addition to the optimal solution. We adapt this algorithm to other scenarios, where the initial set I of infected nodes is known and the network under concern is restricted to at most T-hops away from I. We further propose a community-based algorithm which returns efficiently a good selection of nodes to decontaminate. Finally, we demonstrate the performance of our approaches on three real-world traces.

There are a few open issues that are worth discussing here. First, we assume that once an influential user is presented with good information, he or she would be willing to further spread it to all his or her online friends. In reality, however, it may need to provide extra incentive to these people for combatting against misinformation spreading in online social networks. Second, in this work we assume that both misinformation and good information spread under the same diffusion models. Such an assumption, however, is unnecessary, as in practice, we may want to present the good information in such a way that it spreads faster than misinformation. Third, our work is based on the assumption that the campaign of disseminating good information to combat misinformation is initiated in a centralized manner, say, by the service provider of the online social network. This may not be realistic for some misinformation decontamination campaigns, especially when there is no centralized entity to organize such campaigns, or the service provider of the online social network holds a neutral position on such issues. Despite these realistic challenges, the methods proposed in this work, as well as our rigorous theoretic analysis, shed light on how to develop effective, yet efficient, techniques to contain misinformation spreading in large-scale online social networks.

Acknowledgment

This work is partially supported by the DTRA YIP Grant No. HDTRA1-09-1-0061, and the Los Alamos National Laboratory Directed Research and Development Program, project number 20110093DR.

References

[1] http://articles.cnn.com/2011-05-02/tech/osama.bin.laden.twitter1bin-tweet-twitter-user?s=PM:TECH.

[2] http://www.facebook.com/OccupyWallSt.

[3] www.pcworld.com/businesscenter/article/163920/swineflufrenzydemonstratestwittersachillesheel.html.

[4] http://articles.cnn.com/2011-07-04/tech/fox.hack1tweets-twitter-feed-twitter-users?s=PM:TECH.

[5] http://blog.twitter.com/2011/03/numbers.html.

[6] C. Grier, K. Thomas, V. Paxson, M. Zhang, @spam: The underground on 140 characters or less, in: Proceedings of the 17th ACM Conference on Computer and Communications Security, CCS '10, ACM, New York, NY, USA, 2010, pp. 27–37.

[7] C. Budak, D. Agrawal, A. El Abbadi, Limiting the spread of misinformation in social networks, in: Proceedings of the 20th International Conference on World Wide Web, WWW '11, ACM, New York, NY, USA, 2011, pp. 665–674.

[8] A. Goyal, F. Bonchi, L. Lakshmanan, S. Venkatasubramanian, On minimizing budget and time in influence propagation over social networks, Social Network Analysis and Mining (2012) 1–14.

[9] B. Viswanath, A. Mislove, M. Cha, K.P. Gummadi, On the evolution of user interaction in facebook, in: Proceedings of the 2nd ACM Workshop on Online Social Networks, WOSN '09, ACM, New York, NY, USA, 2009, pp. 37–42.

[10] P. Domingos, M. Richardson, Mining the network value of customers, in: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '01, ACM, New York, NY, USA, 2001, pp. 57–66.

[11] D. Kempe, J. Kleinberg, E. Tardos, Maximizing the spread of influence through a social network, in: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '03, ACM, New York, NY, USA, 2003, pp. 137–146.

[12] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, N. Glance, Cost-effective outbreak detection in networks, in: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '07, ACM, New York, NY, USA, 2007, pp. 420–429.

[13] N. Chen, On the approximability of influence in social networks, in: Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '08, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2008, pp. 1029–1037.

[14] D.M. Nicol, M. Liljenstam, Models and analysis of active worm defense, in: Proceedings of the Third international conference on Mathematical Methods, Models, and Architectures for Computer Network Security, MMM-ACNS'05, Springer-Verlag, Berlin, Heidelberg, 2005, pp. 38–53.

[15] S. Tanachaiwiwat, A. Helmy, Encounter-based worms: analysis and defense, Ad Hoc Network 7 (7) (2009) 1414–1430.

[16] P. Dubey, R. Garg, B. De Meyer, Competing for customers in a social network: the quasi-linear case, in: Proceedings of the Second International Conference on Internet and Network Economics, WINE'06, Springer-Verlag, Berlin, Heidelberg, 2006, pp. 162–173.

[17] S. Bharathi, D. Kempe, M. Salek, Competitive influence maximization in social networks, in: Proceedings of the 3rd International Conference on Internet and Network Economics, WINE'07, Springer-Verlag, Berlin, Heidelberg, 2007, pp. 306–311.

[18] J. Kostka, Y.A. Oswald, R. Wattenhofer, Word of mouth: Rumor dissemination in social networks, in: Proceedings of the 15th International Colloquium on Structural Information and Communication Complexity, SIROCCO '08, Springer-Verlag, Berlin, Heidelberg, 2008, pp. 185–196.

[19] G. Yan, G. Chen, S. Eidenbenz, N. Li, Malware propagation in online social networks: nature, dynamics, and defense implications, in: Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security, ASIACCS '11, ACM, New York, NY, USA, 2011, pp. 196–206.

[20] W. Xu, F. Zhang, S. Zhu, Toward worm detection in online social networks, in: Proceedings of the 26th Annual Computer Security Applications Conference, ACSAC '10, ACM, New York, NY, USA, 2010, pp. 11–20.

[21] W. Chen, C. Wang, Y. Wang, 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, KDD '10, ACM, New York, NY, USA, 2010, pp. 1029–1038.

[22] G.L. Nemhauser, L.A. Wolsey, M.L. Fisher, An analysis of approximations for maximizing submodular set functions, Mathematical Programming 14 (1978) 265–294.

[23] M. Kimura, K. Saito, R. Nakano, Extracting influential nodes for information diffusion on a social network, in: Proceedings of the 22nd national conference on Artificial intelligence – AAAI'07, vol. 2, AAAI Press, 2007, pp. 1371–1376.

[24] T.N. Dinh, D.T. Nguyen, M.T. Thai, Cheap, easy, and massively effective viral marketing in social networks: truth or fiction?, in: Proceedings of the 23rd ACM Conference on Hypertext and Social Media, HT '12, ACM, New York, NY, USA, 2012, pp 165–174.

[25] R. Raz, S. Safra, A Sub-Constant Error-Probability Low-Degree Test, and a Sub-Constant Error-Probability PCP Characterization of NP, STOC.

[26] W. Chen, Y. Yuan, L. Zhang, Scalable influence maximization in social networks under the linear threshold model, in: Proceedings of the 2010 IEEE International Conference on Data Mining, ICDM '10, IEEE Computer Society, Washington, DC, USA, 2010, pp. 88–97.

[27] S. Fortunato, Community detection in graphs, Physics Reports 486 (35) (2010) 75–174.

[28] T.N. Dinh, Y. Xuan, M.T. Thai, Towards social-aware routing in dynamic communication networks, in: Proceedings of the 28th International Performance Computing and Communications Conference, IPCCC '09, 2009, pp. 161–168.

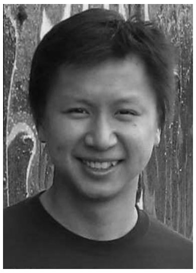
[29] N.P. Nguyen, T.N. Dinh, Y. Xuan, M.T. Thai, Adaptive algorithms for detecting community structure in dynamic social networks, in: Proceedings of the 31th International Conference on Computer Communications, Infocom '11, 2011, pp. 2282–2290.

[30] N.P. Nguyen, T.N. Dinh, S. Tokala, M.T. Thai, Overlapping communities in dynamic networks: their detection and mobile applications, in: Proceedings of the 17th Annual International Conference on Mobile Computing and Networking, MobiCom '11, ACM, New York, NY, USA, 2011, pp. 85–96.

[31] V.D. Blondel, J.-L. Guillaume, R. Lambiotte, E. Lefebvre, Fast unfolding of communities in large networks, Journal of Statistical Mechanics: Theory and Experiment 2008 (10) (2008) P10008.

[32] S. Fortunato, A. Lancichinetti, Community detection algorithms: a comparative analysis, in: Proceedings of the Fourth International ICST Conference on Performance Evaluation Methodologies and Tools, VALUETOOLS '09, ICST, ICST, Brussels, Belgium, 2009, pp. 27:1–27:2.

[33] M.A. Porter, J.-P. Onnela, P.J. Mucha, Communities in networks, Notices of the AMS 56 (9).



Nam P. Nguyen received his B.S. degree from Vietnam National University, VN (2007) and M.S. degree from Ohio University, USA (2009) both in Applied Mathematics. He is now a Ph.D. candidate in Computer Science at University of Florida, USA. His interests include cyber-security, complex networks structural analysis and vulnerability assessment, as well as effective approximation algorithms for practical mobile and social networking problems.

Guanhua Yan obtained his Ph.D. degree in Computer Science from Dartmouth College, USA, in 2005. From 2003 to 2005, he was a visiting graduate student at the Coordinated Science Laboratory in the University of Illinois at Urbana-Champaign. He is now a Technical Staff Member in the Information Sciences Group (CCS-3) at the Los Alamos National Laboratory. His research interests are cyber-security, networking, and large-scale modeling and simulation techniques. He has contributed about 50 articles in these fields.

My T. Thai received her Ph.D. degree in computer science from the University of Minnesota, Twin Cities, in 2006. She is an associate professor in the CISE Department, University of Florida. Her current research interests include algorithms and optimization on network science and engineering. She also serves as an associate editor for the Journal of Combinatorial Optimization (JOCO) and Optimization Letters and a conference chair of COCOON 2010 and several workshops in an area of network science. She is a recipient of DoD Young Investigator Awards and NSF CAREER awards. She is a member of the IEEE.