Targeted Protection Maximization in Social Networks

Jianxiong Guo, Yi Li, Weili Wu *Member, IEEE*

Abstract—Even though the widespread use of social platforms provides convenience to our daily life, it causes some bad results at the same time. For example, misinformation and personal attack can be spread easily on social networks, which drives us to study how to block the spread of misinformation effectively. Unlike the classical rumor blocking problem, we study how to protect the targeted users from being influenced by rumor, called targeted protection maximization (TPM). It aims to block the least edges such that the expected ratio of nodes in targeted set influenced by rumor is at most β. Under the IC-model, the objective function of TPM is monotone non-decreasing, but not submodular and not supermodular, which makes it difficult for us to solve it by existing algorithms. In this paper, we propose two efficient techniques to solve TPM problem, called Greedy and General-TIM. The Greedy uses simple Hill-Climbing strategy, and get a theoretical bound, but the time complexity is hard to accept. The second algorithm, General-TIM, is formed by means of randomized sampling by Reverse Shortest Path (Random-RS-Path), which reduces the time consuming significantly. A precise approximation ratio cannot be promised in General-TIM, but in fact, it can get good results in reality. Considering the community structure in networks, both Greedy and General-TIM can be improved after removing unrelated communities. Finally, the effectiveness and efficiency of our algorithms is evaluated on several real datasets.

Index Terms—Targeted Protection Maximization, Rumor Blocking, Social Network, Randomized Algorithm

1 Introduction

The online social media, such as Facebook, Twitter, Flickr, Google++ and Linkedln, was booming rapidly in last decades, where billion of people communicated with each other and produce a lot of information at any time. The opportunities were provided by the applications of online social networks (OSNs) for fast information propagation. Even that, A platform provided by the OSNs to misinformation conveniently. Misinformation in OSNs can be a rumor, a piece of fake news, or information generated due to misunderstanding, which causes severe consequences and even panics. For example, the rumor "Barack Obama is injured" spread in Twitter in 2013 made US stock market crash immediately and the fake news about Hillary Clinton selling weapons to ISIS spread very fast in Facebook in 2016 [1], which damaged her election. A very recent example occurred on October 28, 2018. A bus crashed into a car and then fell down into deep water, which costed 15 lives. In OSNs, all comments blamed the car driver because from photos on the news someone analyzed that it was car's fault, which got the car driver arrested. However, through the police's investigation, the car driver was found innocent and the accident was actually caused due to a fight between the bus driver and a passenger.

In social networks, information or influence spread from node to node via cascades, which are activated by a set of seeds. The formal study of information propagation in social networks was begun from Kempe et al. [2] where the influence maximization (IM) problem was formulated. Besides, two classical diffusion models, accepted by most researchers, was proposed: independent cascade model (IC-model) and linear threshold model (LT-model). In this paper, we use IC-model as our fundamental model, the details will be described in Sec. 3. In most existing methods, the monotonicity and submodularity of objective function need to be used. If the objective function is not submodular, it is a challenge how to solve it with a theoretical bound and low time complexity.

Rumor Blocking: The problem of rumor blocking is studied intensively before. Budak et al. [3] proposed the problem of rumor blocking firstly, which is formulated as a combinatorial optimization problem. In addition, the NP-hardness was proved and based on its submodularity, the approximation ratio for a greedy method is provided in [3]. Existing works on rumor blocking in social networks can be classified into three categories roughly:

1) Removing or protecting nodes which are the most influential so that the spread of misinformation is minimized. [4] [5] [6]
2) Removing a certain number of edges that play an important role in networks to limit spread of misinformation, such as these bridge edges connecting different community. [7] [8] [9]
3) Spreading positive information to compete with misinformation, such that most nodes are influenced by positive information. [3] [10] [11] [12]

Unlike the above works, we propose a new problem, called Targeted Protection Maximization (TPM). The main difference between TPM and classical rumor blocking is that we only consider targeted set. The TPM can be described roughly as to block (remove) an edge set of least edges such that the nodes in targeted set are protected from influenced

• *J. Guo, Y. Li, W. Wu are with the Department of Computer Science, Erik Jonsson School of Engineering and Computer Science, Univerity of Texas at Dallas, Richardson, TX, 75080 USA*
E-mail: jianxiong.guo@utdallas.edu

Manuscript received April 19, 2005; revised August 26, 2015.

by rumor cascading. This problem can be applied to a lot of real circumstances. For example, cyberbullying, it aims to attack, humiliate or disparage some targeted persons over network, via broadcasting, posting, or sending negative, harmful information of the targets through cyber-media, such as social networks or web forums; or the Internet administration of a government want to protect the adolescents in their country from toughing pornographic and violent information on the Internet; or a company do not want their customers and related groups to be influenced by its adverse companies.

In this paper, we show that if there are a small number of relationships (edges) removed from social networks, the misinformation to the targeted victims can be minimized. Based on this idea, the objective function $h_G(\cdot)$ of TPM problem will be formulated, and we prove that it is a monotone non-decreasing, but not submodular and supermodular function. However, the standard Greedy algorithm often obtains good results on a large number of non-submodular applications, which will be described detailedly in Sec. 3. Even if Greedy algorithm is simple and effective, the objective function is hard to compute due to computing the expected influence is #P-hard [13]. Monte Carlo simulation is a common method to estimate expected value, but the computational cost is too expensive. In order to overcome this shortcoming, randomized algorithm based on sampling popped up gradually [14] [15] [16]. The idea of reserve influence sampling (RIS) for the IM problem was proposed firstly by Brogs et al. [14]. Inspired by this idea, we proposed a novel and effective sampling method, which based on the concept of random Reverse shortest path (Random-RS-Path). The Random-RS-Path is a random shortest path from a node in rumor set to targeted set. Based on the concept of Random-RS-Path, we propose a randomized algorithm, called General-TIM. As we known, it is impossible to speed up an algorithm significantly without lowering the performance bound in the most application. Despite that this sampling method, Random-RS-Path, cannot be promised to give an unbiased estimation, it can give us an acceptable result which is very close to Greedy algorithm, which will be evaluated in real dataset later. Besides, we propose a Greedy algorithm based on community structure of real-world networks where some unnecessary communities will not be considered. After removing unrelated community, both Greedy and General-TIM algorithm can be improved significantly. Our contribution in this paper are summarized as follows:

- This is the first attempt to study rumor blocking problem for targeted set under the IC-model in social networks. Then, the problem of Targeted Protection Maximization (TPM) is formulated.
- We prove that TPM problem under the IC-model is monotone non-decreasing, but not submodular and not supermodular.
- We propose Greedy algorithm to solve TPM problem, and prove an upper bound for optimal solution. Besides, we develop a new sampling method based on Reverse Shortest Path (Random-RS-Path), which is a valid estimation for the objective function of TPM problem. Then, we propose General-TIM algorithm.
- Our algorithms are evaluated on two real-world datasets. The results show both Greedy and General-TIM are better than heuristics, and General-TIM is a good estimation for TPM problem.
- We propose a speedup method for Greedy algorithm based on community structure in social networks.

Related Works: Budak et al. [3] was the first to study the problem of rumor blocking, and showed that rumor blocking problem is a submodular maximization problem under the competitive model. He et al. [17] showed a $(1-1/e)$-approximation algorithm for the competitive linear threshold for the problem of rumor influence minimization. Fan et al. proposed the problem of least cost rumor blocking, and proved a $(1-1/e)$-approximation algorithm under the opportunistic one-active-one model. Later, Nguyen et al. [18] presented the IT-Node Protector problem, which removed the nodes with high influence to block the spread of rumor. If you want to learn more about the problem of misinformation, please read a survey [19] about false information, which is written by Srijan et al. However, we have to require Monte Carlo (MC) simulation to compute the expected influence given a seed set, and its computational cost is too high to apply to large real-world networks, even if there exists some effective methods to improve the utility of MC simulation. Leskovec et al. proposed an improved method called CELF [20], which estimate the upper bounds of influence function because of its submodularity. Most nodes with few influences will not be considered in the later iteration. CELF++, proposed by [21], improves CELF to get better time complexity. Although there are ways to improve MC, it is difficult to achieve the desired effect. TIM/TIM+ [15] and IMM [16] occurred, which makes the IM being scalable under the premise of guaranteeing the approximate ratio. These methods are based on a RIS, proposed by Borgs et al. [14], and determine the number of RR-sets needed to ensure approximation ratio. It required OPT, the optimal expected influence of valid seed set, to estimate the number of reverse reachable set (RR-sets). However, OPT is difficult to determine, [15] [22] proposed a bunch of parameter estimation technique to estimate OPT. Then, IMM appeared, which uses a martingale analysis to estimate OPT more efficiently [16]. This better parameter estimation improves TIM/TIM+.

Organiztion: Sec. 2 descirbes background knowledges and problem formulation. Sec. 3 presents the algorithms for TPM problem. Sec. 4 discusses experimental setup and experimental results. Sec. 5 introduces the improved algorithm based on community structure and Sec. 6 is conclusion.

2 Problem Formulation

In this section, we give the preliminaries, including influence model and notation to this paper, then the problem is formulated.

2.1 Influence Model

A social network can be expressed as a directed graph $G = (V, E)$, usually the users are denoted as V and edge $e = (u, v) \in E$ denotes the relationship between user u and user v. The number of nodes and edges in graph G are n and

m respectively. For a directed edge $e = (u, v)$, u (resp. v) is the incoming (resp. outgoing) neighbor of v (resp. u). The set of incoming neighbors and outgoing neighbors of node v are denoted as $N^-(v)$ and $N^+(v)$ respectively. Let node set and edge set in the directed graph G be denoted as $V(G)$ and $E(G)$ respectively. In order to represent the spreading of new information or technology, Kempe et al. [2] proposed two classical diffusion model, IC-model and LT-model. The process of influence diffusion stops when no new nodes can be activated in this round.

Definition 1 (IC-model). *It assumes that each node v is attempted to be activated independently by its incoming neighbors $N^-(v)$ with activation probability is $p_{uv}, u \in N^-(v)$. Given an activation probability p_{uv} for each pair of edges (u, v), the propagation process can be described in discrete rounds: In round t, each node u activated in round $t-1$ will attempt to activate the nodes in its outgoing neighbors $N^+(u)$, which is inactive in round t, with activation probability p_{uv}. It is worth noting that each node has only one opportunity to make their inactive outgoing neighbors active.*

Definition 2 (LT-model). *It assumes that for each edge $e = (u, v) \in E(G)$, a weight b_{uv} is correlated with it. Each node $v \in V(G)$ satisfies that $\Sigma_{u \in N^-(v)} b_{uv} \leq 1$. Besides, Each node $v \in V(G)$ is correlated with a threshold λ_v, which is uniformly distributed in interval $[0, 1]$. Given that, the propagation process can be described in discrete rounds: In round t, the nodes that have been activated in round $t-1$ are still active. Any inactive node v will become active if the total weight associated with active nodes in its incoming neighbors $N^-(v)$ are greater than λ_v.*

Then, we need to define a monotone and submodular function. A set function $f : 2^V \to \mathbb{R}$ is monotone iff $f(S) \leq f(T)$ for any $S \subseteq T \subseteq V$. A set function is submodular iff $f(S \cup \{u\}) - f(S) \geq f(T \cup \{u\}) - f(T)$ for any $S \subseteq T \subseteq V$ and $u \notin V \setminus T$. If we can know that a function has the property of monotonicity and submodularity, we can optimize it easily with the help of existing theory, such as a $(1 - 1/e)$-approximation obtained by classical hill-climbing algorithm [23].

In this paper, the IC-model will be adopted by us as fundamental influence model to solve our problem, but we will compare the difference of this problem under IC-model and LT-model.

2.2 Realization

Given a directed graph $G = (V, E)$, a realization g is a subgraph of G where $V(\mathrm{g}) = V(G)$ and $E(\mathrm{g}) \subseteq E(G)$. Under the IC-model, the diffusion probability of a realization g is equal to 1. For each edge $e = (u, v) \in E(G)$, it appears in realization g with probability p_{uv}. These edges appear in realization g are referred as to "live" edges. Let $\mathcal{G}$ be the set of all realizations generated from G and $\Pr[\mathrm{g}]$ be the probability of realization g, we have

$$\Pr[\mathrm{g}] = \prod_{e \in E(\mathrm{g})} p_e \prod_{e \in E(G) \setminus E(\mathrm{g})} (1 - p_e) \tag{1}$$

Obviously, there are 2^m possible realization altogether. The propagation in a realization g is a deterministic process. Thus, we can think about the propagation process from two different perspectives. Given a seed set S, the propagation can be considered as a stochastic propagation process on graph G with a probability distribution, or a deterministic propagation process on a realization g generated from G with a probability distribution.

Under the LT-model, a realization g of G is generated differently. Given a graph $G = (V, E)$, for each node $v \in E(G)$, at most one of its incoming edges can be selected as edge in $E(\mathrm{g})$. For each node $u \in N^-(v)$, edge (u, v) is selected with probability b_{uv} and no edge from $N^-(v)$ is selected with probablity $1 - \Sigma_{u \in N^-(v)} b_{uv}$. These edges appear in realization g are referred as to "live" edges. It is worth noting that for any node u and v, there is at most one possible path connecting them in a realization g of G, which will be useful later.

2.3 Probem Definition

Given the negative (rumor) node set S and targeted node set T, we can define $f_G(S, T)$ as the expected number of nodes in T that are influenced by negative information from S. The influence from S to T in graph G under the IC-model can be defined as follows:

$$f_G(S, T) = \sum_{\mathrm{g} \in \mathcal{G}} \Pr[\mathrm{g}] \cdot f_{\mathrm{g}}(S, T) \tag{2}$$

where $f_{\mathrm{g}}(S, T)$ is the number of nodes in T can be reached from any node in S in the realization g of graph G. From above, the Targeted Protection Maximization (TPM) problem can be defined as follows:

Definition 3 (Targeted Protection Maximization). *Given a social network $G = (V, E)$, a set of negative (rumor) nodes $S \subseteq V(G)$, a set of targeted nodes $T \subseteq V(G) \setminus S$, and a threshold $\beta \in [0, 1]$, the TPM problem aims to block (remove) a edge set $I \subseteq E(G)$ of least edges such that the expected ratio of nodes in T influenced by rumor on the network $G(I)$, is at most β of that on the network G.*

Formally, TPM can be express as $\min\{|I| : I \subseteq E(G)\}$ such that $f_{G(I)}(S, T) \leq \beta \cdot f_G(S, T)$. Here, $|I|$ is the cardinality of a set I. For an edge $e \in E(G)$, we define $G(e)$ as the graph $(V, E \setminus e)$, so $G(D)$ is the graph $(V, E \setminus D)$ for $D \subseteq E(G)$. Here, we define $h_G(I) = f_G(S, T) - f_{G(I)}(S, T)$. This problem is equivalent to select an edge set of least edges such that

$$h_G(I) \geq (1 - \beta) \cdot f_G(S, T) \tag{3}$$

given two sets S and T. The TPM problem is NP-hard under the IC-model and LT-model. As in following theorem, we prove that the objective function $h_G(\cdot)$ is monotone non-decreasing, but not submodular and supermodular under the IC-model.

Theorem 1. *The objective function $h_G(\cdot)$ is monotone non-decreasing under the IC-model.*

Proof. Given directed graph $G = (V, E)$, and edge set I and an edge $e = (u, v) \in E \setminus I$, we need to prove that $f_{G(I)}(S, T)$ is monotone non-increasing, which can be expressed as the form $f_{G(I)}(S, T) - f_{G(I \cup \{e\})} \geq 0$. In other words, we have $f_{G(I)}(S, T) - f_{G(I \cup \{e\})} = \sum_{\mathrm{g} \in \mathcal{G}(I)} \Pr[\mathrm{g} | \mathrm{g} \in \mathcal{G}(I)] \cdot f_g(S, T) - \sum_{\mathrm{g} \in \mathcal{G}(I \cup \{e\})} \Pr[\mathrm{g} | \mathrm{g} \in \mathcal{G}(I \cup \{e\})] \cdot f_{\mathrm{g}}(S, T)$, where $\Pr[\mathrm{g} | \mathrm{g} \in \mathcal{G}(I)]$ stand for the probability of realization g in graph $G(I)$.

For any realization $\mathrm{g} = (V, E_{\mathrm{g}})$ in $\mathcal{G}(I \cup \{e\})$, there is a one-to-two mapping from $\mathcal{G}(I \cup \{e\})$ to $\mathcal{G}(I)$. In other words, given $\mathrm{g} \in \mathcal{G}(I \cup \{e\})$, there are only two corresponding realization g' and g'' in $\mathcal{G}(I)$, where $\mathrm{g}' = \mathrm{g} = (V, E_{\mathrm{g}})$ and $\mathrm{g}'' = (V, E_{\mathrm{g}} \cup \{e\})$ respectively. Thus, the probability of realization $\mathrm{g}' \cup \mathrm{g}''$ in graph $G(I)$ is

$$\begin{aligned}
&\Pr[\mathrm{g}' \cup \mathrm{g}''|\mathrm{g}', \mathrm{g}'' \in \mathcal{G}(I)] \\
&= \Pr[\mathrm{g}'|\mathrm{g}' \in \mathcal{G}(I)] + \Pr[\mathrm{g}''|\mathrm{g}'' \in \mathcal{G}(I)] \\
&= \Pr[\mathrm{g}'|\mathrm{g}' \in \mathcal{G}(I \cup \{e\})](1 - p_e) + \Pr[\mathrm{g}'|\mathrm{g}' \in \mathcal{G}(I \cup \{e\})]p_e \\
&= \Pr[\mathrm{g}'|\mathrm{g}' \in \mathcal{G}(I \cup \{e\})]
\end{aligned}$$

Then, the expectation influence $f_{\mathrm{g}' \cup \mathrm{g}''}(S, T)$ in g' and g'' is $f_{\mathrm{g}' \cup \mathrm{g}''}(S, T) = (1 - p_e) \cdot f_{\mathrm{g}'}(S, T) + p_e \cdot f_{\mathrm{g}''}(S, T) \geq f_{\mathrm{g}}(S, T)$ because of $f_{\mathrm{g}''}(S, T) \geq f_{\mathrm{g}}(S, T)$. Therefore, $f_{G(I)}(S, T) - f_{G(I \cup \{e\})} \geq 0$, which completes the proof. □

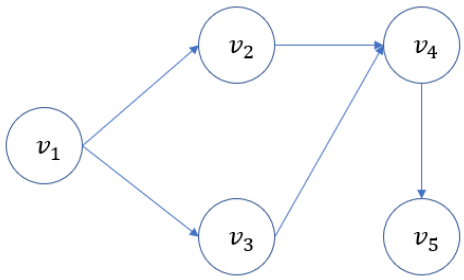


Fig. 1. A realization g of G to show $h_G(\cdot)$ is not submodular under the IC-model

Theorem 2. *The objective function $h_G(\cdot)$ is not submodular under the IC-model.*

Proof. We prove by a counterexample. Shown as Fig. 1, we consider a realization $\mathrm{g} = (V, E)$, $V = \{v_1, v_2, v_3, v_4, v_5\}$ and $E = \{(v_1, v_2), (v_1, v_3), (v_2, v_4), (v_3, v_4), (v_4, v_5)\}$, then let $S = \{v_1\}$ and $T = \{v_2, v_3, v_4, v_5\}$, $I_1 = \emptyset$ and $I_2 = \{(v_1, v_2)\}$. Here, we have $h_{\mathrm{g}}(I_1) = 0$ and $h_{\mathrm{g}}(I_2) = 1$. Putting edge $e = (v_1, v_3)$ into I_1 and I_2, we have $h_{\mathrm{g}}(I_1 \cup \{e\}) = 1$ and $h_{\mathrm{g}}(I_2 \cup \{e\}) = 4$. Then, $h_{\mathrm{g}}(I_1 \cup \{e\}) - h_{\mathrm{g}}(I_1) = 1 < h_{\mathrm{g}}(I_2 \cup \{e\}) - h_{\mathrm{g}}(I_2) = 3$ when $I_1 \subseteq I_2$. Thus, $h_{\mathrm{g}}(\cdot)$ is not a submodular function. □

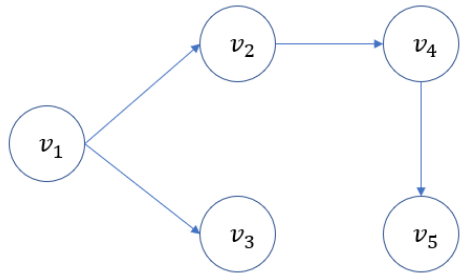


Fig. 2. A realization g of G to show $h_G(\cdot)$ is not supermodular under the IC-model

Theorem 3. *The objective function $h_G(\cdot)$ is not supermodular under the IC-model.*

Proof. We prove by a counterexample. Shown as Fig. 2, we consider a realization $\mathrm{g} = (V, E)$, $V = \{v_1, v_2, v_3, v_4, v_5\}$ and $E = \{(v_1, v_2), (v_1, v_3), (v_2, v_4), (v_4, v_5)\}$, then let $S = \{v_1\}$ and $T = \{v_2, v_3, v_4, v_5\}$, $I_1 = \emptyset$ and $I_2 = \{(v_4, v_5)\}$. Here, we have $h_{\mathrm{g}}(I_1) = 0$ and $h_{\mathrm{g}}(I_2) = 1$. Putting edge $e = (v_1, v_2)$ into I_1 and I_2, we have $h_{\mathrm{g}}(I_1 \cup \{e\}) = 3$ and $h_{\mathrm{g}}(I_2 \cup \{e\}) = 3$. Then, $h_{\mathrm{g}}(I_1 \cup \{e\}) - h_{\mathrm{g}}(I_1) = 3 > h_{\mathrm{g}}(I_2 \cup \{e\}) - h_{\mathrm{g}}(I_2) = 2$ when $I_1 \subseteq I_2$. Thus, $h_{\mathrm{g}}(\cdot)$ is not a supermodular function. □

Consider the influence maximization (IM) problem: Given a graph $G = (V, E)$ and a positive integer k, IM select a seed set S of k nodes to maximize the expected spread of influence $\sigma(\cdot)$. In general, a set function $f(\cdot)$ over k-cardinality constraint can obtain a $(1 - 1/e)$-approximation ratio, i.e. $f(S) \geq (1 - 1/e) \cdot f(S^*)$, if $f(\cdot)$ is a monotone non-decreasing submodular function. However, our objective $h_G(\cdot)$ is not submodular and supermodular and thus, the $(1 - 1/e)$-approximation ratio cannot be hold with k-cardinality constraint. However, under the LT-model, the situation is different from that under the IC-model:

Theorem 4. *The objective function $h_G(\cdot)$ is monotone non-decreasing, submodular under the LT-model.*

Proof. The proof of monitonicity of $h_G(\cdot)$ can be extended from [24]. Next, in order to prove $h_G(\cdot)$ is submodualr, we need to prove each realization $h_{\mathrm{g}}(\cdot)$ is submodular. From above, we have known that there is at most one possible path connecting each node in S to each node in T. To see this, let I_1 and I_2 be two edge set such that $I_1 \subseteq I_2$ and g is a realization G under the IT-model, then we consider the value of $f_{\mathrm{g}(I_1)}(S, T) - f_{\mathrm{g}(I_1 \cup \{e\})}(S, T)$, which is equal to $h_{\mathrm{g}}(I_1 \cup \{e\}) - h_{\mathrm{g}}(I_1)$. For any node $v \in T$, because there is at most one path from node $u \in S$ to v, it cannot be influence by S if and only if all the paths from each $u \in S$ to v have been blocked or such a path does not exist. When adding a new edge e to I_1, the number of nodes in T cannot be influenced by S, but influenced before, which is at least as large as the number of new immune nodes in T when adding this new e to I_2. Obviously, we can get that $h_{\mathrm{g}}(I_1 \cup \{e\}) - h_{\mathrm{g}}(I_1) \geq h_{\mathrm{g}}(I_2 \cup \{e\}) - h_{\mathrm{g}}(I_2)$ when $I_1 \subseteq I_2$. Besides, from equation (2), $h_G(\cdot)$ is submodular, which completes the proof. □

The methods suitable to a monotone non-decreasing, submodular function can be applied to solve $h_G(\cdot)$ under the LT-model. There is an $(1 - 1/e)$-approximation by standard Greedy algorithm since $h_G(\cdot)$ is monotone non-decreasing and submodular. Under the LT-model, the problem similar to TPM problem has been solve before [25]. Despite this, we want to know what will happen when standard Greedy algorithm is applied to solve $h_G(\cdot)$ under the IC-model.

In practice, as a matter of experience, the standard Greedy algorithm often obtains good results on a large number of non-submodular applications. In order to theorize for the empirical success of standard Greedy algorithm on non-submodular function, [26], [27], [28] used curvature α to quantify the closeness between submodular function and modular function. Later, Das et al. [29] proposed the modularity ratio γ to quantify the closeness between a set function and submodular function. Bian et al. [30] prove that given curvature $\alpha \in [0, 1]$ and submodularity ratio $\gamma \in$

$[0, 1]$, the Greedy algorithm can obtain a $(1/\alpha)(1 - e^{-\gamma\alpha})$-approximation for maximizing non-decreasing set function with cardinality constraint.

3 Solution For TPM

From above, we know that the standard Greedy algorithm often obtains good results on some non-submodular problem, which means that some methods suitable to the submodular problem can be applied to non-submodular problem as well. Therefore, we propose following methods to solve TPM problem.

3.1 Greedy Algorithm

Now, we introduce the first technique, call Greedy, to solve the TPM problem. The main idea is to use Hill-Climbing algorithm to block the edge that has a maximum marginal gain. At each step, it selects an edge e from $E\backslash I$ such that adding e to I maximizes $h_G(I \cup \{e\}) - h_G(I)$. We repeat this until the fraction of nodes influenced by rumor S is less than threshold β. The pseudo-code of Greedy algorithm is shown as follows:

Algorithm 1 Greedy (G, S, T, β)

Input: $G = (V, E)$, S, T and $\beta \in [0, 1]$
Output: $I \in E$ satisfies $f_{G(I)}(S, T) \leq \beta \cdot f_G(S, T)$
1: Initial $I \leftarrow \emptyset$
2: Compute $f_G(S, T)$, and $K \leftarrow f_G(S, T)$
3: **while** $f_{G(I)}(S, T) > \beta K$ **do**
4: Select an edge e from $E\backslash I$ such that maximizing $h_G(I \cup \{e\}) - h_G(I)$
5: $I \leftarrow I \cup \{e\}$
6: **end while**
7: **return** edge set I

Here, I is the set of blocked edges, and stand for the approximate solution returned by Algorithm 1. From last section, $h_G(I)$ is a $(1/\alpha)(1 - e^{-\gamma\alpha})$-approximation solution with cardinality $|I|$ constraint, provided that curvature α and modularity ratio γ exist. In other words, $h_G(I) \geq (1/\alpha)(1 - e^{-\gamma\alpha}) \cdot h_G(I^*)$, where I^* is the edge set with cardinality $|I|$ that maximize the value of $h_G(\cdot)$. Because of this, we can prove that the solution return by Algorithm 1 will not be far from the optimal solution.

Theorem 5. *The edge set I, $|I| = d$, returned by Algorithm 1 for the TPM problem, here n satisfies the following inequality,*

$$d \leq |OPT| + \max\left\{0, \frac{h_G(I_{d-1})}{\lambda} - \frac{\mu K}{\lambda} + 1\right\} \quad (4)$$

where I_i is the edge set generated at the i^{th} iteration, $K = f_G(S, T)$, $\lambda = \min_{i=1,2,...,d-1}\{h_G(I_{i+1}) - h_G(I_i)\}$, $\mu = (1/\alpha)(1 - e^{-\gamma\alpha}) \cdot (1 - \beta)$ and $|OPT|$ is the size of optimal solution.

Proof. First, we need to show the notation used later. Let $I_i \in \{I_1, I_2, ..., I_n\}$ be the edge sets generated at the i^{th} iteration of Algorithm 1. For any iteration i, let $\max(i)$ be the maximum value of $h_G(I)$ where $I \subseteq E$ and $|I| = i$, in other words, $\max(i) = \max_{I\subseteq E,|I|=i} h_G(I)$. We assume that d is not the value of optimal solution. Because if not, $d = |OPT|$. In this case, the value of optimal solution $|OPT| \leq d - 1$, means that $\max(|OPT|) \leq \max(d - 1)$ at the same time. From above, $h_G(I)$ is a $(1/\alpha)(1-e^{-\gamma\alpha})$-approximation solution with cardinality $|I|$ constraint, provided that curvature α and modularity ratio γ exist [30]. In order to obtain the guarantee for the bound of $d - 1 - |OPT|$, we need to get the bound for $h_G(I_{d-1}) - h_G(I_{|OPT|})$ firstly. We have known that $(1/\alpha)(1 - e^{-\gamma\alpha}) \cdot \max(|OPT|) \leq h_G(I_{|OPT|}) \leq h_G(I_{d-1})$. Then, we have

$$\begin{aligned} &h_G(I_{d-1}) - h_G(I_{|OPT|}) \\ &\leq h_G(I_{d-1}) - (1/\alpha)(1 - e^{-\gamma\alpha}) \cdot \max(|OPT|) \\ &\leq h_G(I_{d-1}) - (1/\alpha)(1 - e^{-\gamma\alpha}) \cdot (1 - \beta) \cdot f_G(S, T) \end{aligned}$$

because of $\max(|OPT|) \geq (1 - \beta) \cdot f_G(S, T)$. Here, we have found the upper bound of $h_G(I_{d-1}) - h_G(I_{|OPT|})$.

Next, we need to find the lower bound of $h_G(I_{d-1}) - h_G(I_{|OPT|})$. Obviously, $h_G(I_{d-1}) - h_G(I_{|OPT|}) \geq \lambda \cdot (d - 1 - |OPT|)$ because of $\lambda = \min_{i=1,2,...,d-1}\{h_G(I_{i+1}) - h_G(I_i)\}$. Thus, we have

$$\begin{aligned} &\lambda \cdot (d - 1 - |OPT|) \\ &\leq h_G(I_{d-1}) - h_G(I_{|OPT|}) \\ &\leq h_G(I_{d-1}) - (1/\alpha)(1 - e^{-\gamma\alpha}) \cdot (1 - \beta) \cdot f_G(S, T) \end{aligned}$$

according to the upper bound and lower bound of $h_G(I_{d-1}) - h_G(I_{|OPT|})$. Then, we have $d - |OPT| \leq h_G(I_{d-1})/\lambda - \mu \cdot f_G(S, T)/\lambda + 1$. Besides, it is possible that $h_G(I_{d-1})/\lambda - \mu \cdot f_G(S, T)/\lambda + 1 < 0$ under certain circumstance. Therefore, $d \leq |OPT| + \max\{0, h_G(I_{d-1})/\lambda - \mu \cdot f_G(S, T)/\lambda + 1\}$, which completes the proof. □

However, we need to calculate $f_{G(I\cup\{e\})}(S, T)$, the expected number of nodes influenced by rumor in T in graph $G(I \cup \{e\})$, for each edge e in each iteration. The time complexity of Greedy algorithm is $O(|I|mnr)$, here r is simulation times using Monte Carlo method. However, this method is difficult to extend to large real networks, because the computational cost is extremely high, even if some improved algorithm, such as CELF and CELF++, exist. This drives us to design a more desirable approach which can obtain a similar solution set in a timely manner.

3.2 General-TIM Algorithm

For influence maximization, Tang et al. [15] proposed the Two-phase Influence Maximization (TIM) algorithm that produces a $(1-1/e-\varepsilon)$-approximation with at least $(1-n^{-\ell})$ probability in $O((k + \ell)(m + n) \log n \cdot \varepsilon^{-2})$. It is based on a technique called reverse influence sampling (RIS). First, we need to introduce two important concepts, reverse reachable set (RR-set) and random RR-set, proposed by Borgs et al. [14]. Given a realization g of graph G and a node v in g, the RR-set is a set in which all the nodes in g can reach v. The random RR-set is a RR-set generated on a realization g sampled from the distribution of realization, like equation (1), and a node which is selected from V randomly. In TIM algorithm, we need to obtain a certain number of random RR-sets. Then, in the problem of influence maximization, for a given node, if it appears more times in these random RR-sets, this node has larger influence than others with high probability. Thus, a maximum coverage problem has been

Algorithm 2 RR-Tree (G, v, g, S, T)

Input: $G = (V, E)$, v, g S, T
Output: $t_{\mathrm{g}}(v)$ and rumor
Initialize $V' \leftarrow \emptyset$ and $E' \leftarrow \emptyset$
Initialize rumor $\leftarrow \emptyset$
Initialize an empty queue Q
$Q \leftarrow Q \cup \{v\}$
while $Q \neq \emptyset$ **do**
  $u \leftarrow Q.pop()$
  children $\leftarrow$ A list of incoming neighbors of u in g, but the order is shuffled
  for w in children **do**
    if $w \notin V'$ **then**
      $Q \leftarrow Q \cup \{w\}$
      $E' \leftarrow E' \cup \{(w, u)\}$
      if $w \in S$ **then**
        flag $\leftarrow$ false
        rumor $\leftarrow \{w\}$
        break
      end if
    end if
  end for
end while
$t_{\mathrm{g}}(v) \leftarrow (V', E')$
return RR-Tree $t_{\mathrm{g}}(v)$ and rumor

formed of selecting at most k nodes such that the most random RR-sets are covered. The TIM algorithm returns a $(1 - 1/e - \varepsilon)$-approximation with at least $(1 - n^{-\ell})$ probability when the θ, the number of random RR-sets, satisfies [15]

$$\theta \geq (8 + 2\varepsilon)n \cdot \frac{\ell \log n + \log \binom{n}{k} + \log 2}{\mathrm{OPT} \cdot \varepsilon^2} \tag{5}$$

However, our protection maximization problem is different from influence maximization, we need to select edge instead of node to cover random RR-sets.

The random RR-set is a node set that is reachable to a random node v. However, in our TPM problem, we need to find an edge such that we can make the nodes in this RR-set unreachable to v by blocking it. It seems very difficult to achieve this goal by means of blocking only one edge, because there are plenty of different paths from any node in RR-set to node v, especially in IC-model. In IC-model, the edge set $E(\mathrm{g})$ of realization g is an arbitrary subset of edge set $E(G)$. Thus, We cannot prevent node v from influenced by its random RR-set by removing an edge. But in LT-model, it is feasible because each node has at most one incoming edge, so there is at most one possible path connecting any node to v in a realization. Fortunately, in our problem, we only need to consider the reachability from nodes in S to nodes in T, which give us some idea to solve this problem approximately.

Considering two nodes u and v, if v can be reached from u in a realization g of G under the IC-model, it is possible that there exist many different paths from u to v and the paths are cyclic. It is not promised that v can be protected by removing only one edge, shown as above. We need to find an effective method to approximate that the node v is protected with high probability by removing one edge. Thus, we propose a new model, shortest path model. Here, we only consider the shortest path from u to v. In other words, we need to select an edge in the shortest path from u to v to be blocked. However, there may be more than one shortest path from u to v, which causes great inconvenience to our handling. Therefore, we can design a new sampling method that can solve it effectively. The definition of Reverse Reachable Tree (RR-Tree) of node $v \in T$ is shown as follows:

Algorithm 3 Random-RS-Path (G, S, T)

Input: $G = (V, E)$, S, T
Output: A RS-Path
Initialize rumor $\leftarrow \emptyset$
Initialize $v \leftarrow \emptyset$
while rumor $= \emptyset$ **do**
  $v \leftarrow$ Select a node from T
  Generate a realization g of G
  $t_{\mathrm{g}}(v)$, rumor $\leftarrow$ RR-Tree (G, v, g, S, T)
end while
RS-Path $\leftarrow$ Path from rumor to v in RR-Tree $t_{\mathrm{g}}(v)$
return RS-Path

Definition 4 (RR-Tree)**.** *Given v be a node in T, and g be a realization of G under the IC-model. The RR-tree for v in g is a reverse breath-first spanning tree of v in g. Here, the spanning tree is random, which means that if a node has multiple outgoing neighbors, we select one from them randomly. The tree stops when the first rumor node appears.*

Given a realization g of G under the IC-model, here node $v \in T$ and RR-tree for v in g, denoted as $t_{\mathrm{g}}(v)$, we can notice that the nodes in $t_{\mathrm{g}}(v)$ is the same as the nodes in RR-set for v in g except root node v. The pseudo-code of RR-Tree algorithm is shown as Algorithm 2. For each node in $t_{\mathrm{g}}(v)$, there is only one path to root node v and the path is acyclic. Thus, if removing an edge in the path from u to v, the influence from u to v in this RR-tree $t_{\mathrm{g}}(v)$ is blocked. In the TPM problem, only paths from $u \in S$ to $v \in T$ needed to be considered. If $v \in T$ cannot be reached from $u \in S$ in a realization g of G, we do not need to remove any edge. There are two possibilities for this to happen. First, there is no path from u to v in graph G. Second, there is no path from u to v in this realization g of G. If there is no path from u to v in many realizations of G, it means that the influence from u to v appears with low probability even though it is not impossible. In addition, if there are more than one shortest path from u to v in g, we need to select one of them uniformly and randomly. Why? For example, there are two shortest path from node $u \in S$ to $v \in T$ in g, $\{(u, w_1), (w_1, v)\}$ and $\{(u, w_2), (w_2, v)\}$ respectively, we connot promise v is protected from u if we only block path $\{(u, w_1), (w_1, v)\}$ whenever it happens. We need these two shortest path to appear as representative for the shortest path with the same probability. Therefore, the shortest path should be selected uniformly and randomly.

From above we have a problem that how to ensure that the shortest path we select is random and uniform when there are more than one shortest path between two nodes? In Algorithm 2, it checks whether a node has been

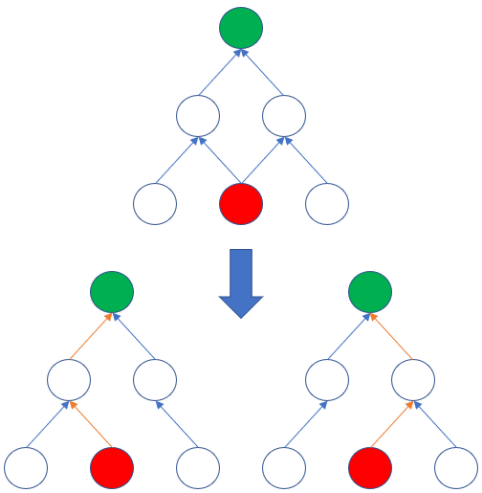


Fig. 3. A sketch to show that we select the shortest path randomly when mutiple shortest path exist

discovered before enqueueing the node rather than delaying this check until the node is dequeued from the queue. When we dequeue a node u from the queue, we need to add all its unvisited child nodes into the queue. In this step, it can be guaranteed that the shortest path we select is random if and only if for each node, the order of adding its unvisited child nodes into the queue is random. Therefore, given $v \in T$, we can find a shortest path randomly from a node in S to v in g by two steps:

1) Generating a RR-Tree $t_{\mathrm{g}}(v)$ and a rumor node.
2) Obtaining the path from this rumor node to v in $t_{\mathrm{g}}(v)$, which is unique and shortest.

Fig. 3 is an example to show the shortest path is arbitrary when there are more than one shortest path. In Fig. 3, rumor node is red and targeted node is green. First, we generated a RR-tree for green node, then obtains a shortest path from red node to green node in this RR-Tree. According to Definition 4, the RR-tree is a random spanning tree, so the two shortest path appear in this RR-tree with the same probability. Until now, the definiton of Random Reverse Shortest Path (Random-RS-Path) can be formulated formally as follows:

Definition 5 (Random-RS-Path)**.** *Let* g *be a random realization of* G *sampling under the IC-model,* v *is a node selected randomly from* T*. The Random-RS-Path is the path from a node* $u \in S$ *to* v *in RR-Tree* $t_g(v)$*, which is unique and shortest.*

Given $G = (V, E)$, rumor set S and targeted set T, the pseudo-code of Random-RS-Path algorithm is shown as Algorithm 3. The RS-path returned by Algorithm 3 is one of path with the fewest edges from a node u in S to another node v in T. The while loop in line 5 exists because it is possible that v cannot encounter a node in S eventually, if it happans, we need to sample again. Back to Algorithm 2, in line 7, the order of adding $u's$ children into the queue is arbitrary as we said before. We need to record the visited edge in line 11, so that want can find the shortest path from first visited rumor to node v in Algorithm 3. Therefore, we can protect v from influenced by u with the high probability by removing an edge in this shortest path. When there are more than one shortest path from u to v, each path in the shortest paths is selected with the same probability. We cannot promise to block all the influence from S, but the estimation is accurate when β is not very small, we can see it in later experiment.

Algorithm 4 General-TIM (G, S, T, θ, β)

Input: $G = (V, E)$, S, T, θ and $\beta \in [0, 1]$
Output: $I \in E$ satisfies $f_{G(I)}(S, T) \leq \beta \cdot f_G(S, T)$

1: Initialize a set $R \leftarrow \emptyset$
2: Initilize $I \leftarrow \emptyset$
3: Generate θ Random-RS-Paths and insert them into R
4: Compute $f_G(S, T)$, and $K \leftarrow f_G(S, T)$
5: **while** $f_{G(I)}(S, T) > \beta K$ **do**
6: Select an edge $e \in E \backslash I$ that cover the most number of Random-RS-Paths in R
7: $I \leftarrow I \cup \{e\}$
8: $R \leftarrow R \backslash \{$Random-RS-Paths covered by $e\}$
9: **end while**
10: **return** edge set I

If the edge, that covers more Random-RS-Paths, is removed, it is possible that more nodes in T are protected. Thus, this problem is transformed to maximum coverage problem of selecting least edges to cover the Random-RS-Paths as many as possible until $f_{G(I)}(S, T) \leq \beta \cdot f_G(S, T)$. The pseudo-code of the randomized algorithm based on Random-RS-Path sampling, General-TIM, is shown in Algorithm 4. First, θ Random-RS-Paths needed to be sampled. The maximum coverage problem is solved by greedy method. Here, we assume that the value of θ is large enough to make sure this estimation is valid.

4 Experiment

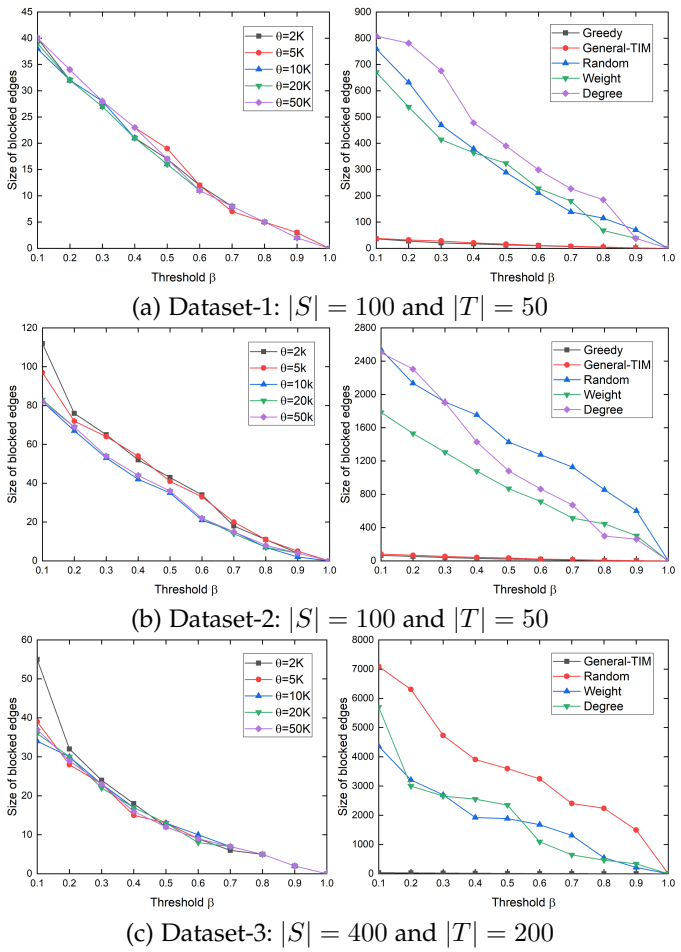
In this section, we show the effectiveness and efficiency of our proposed algorithms on several real social networks. Our goal is to evaluate Algorithm 1 and Algorithm 4 with some common used baseline algorithms.

TABLE 1
The statistics of three datasets

Dataset	n	m	Type	Average degree
Dataset-1	0.4K	1.01K	directed	4
Dataset-2	1.0K	3.15K	directed	6
Dataset-3	6.0K	9.00K	directed	3

4.1 Dataset description and Statistics

Our experiments are based on the dataset from networkrepository.com [31], which is an online network repository. There are three datasets in this experiment. The dataset-1 is a co-authorship network, which is a co-authorship of scientists in network theory and experiments. The dataset-2 is a wiki-vote network, wikipedia who-votes-on-whom network. The dataset-3 is Erdos992's link structure and discover valuable insights using the interactive network data visualization and analytics platform. The statistics information of the two datasets is represented in table 1.

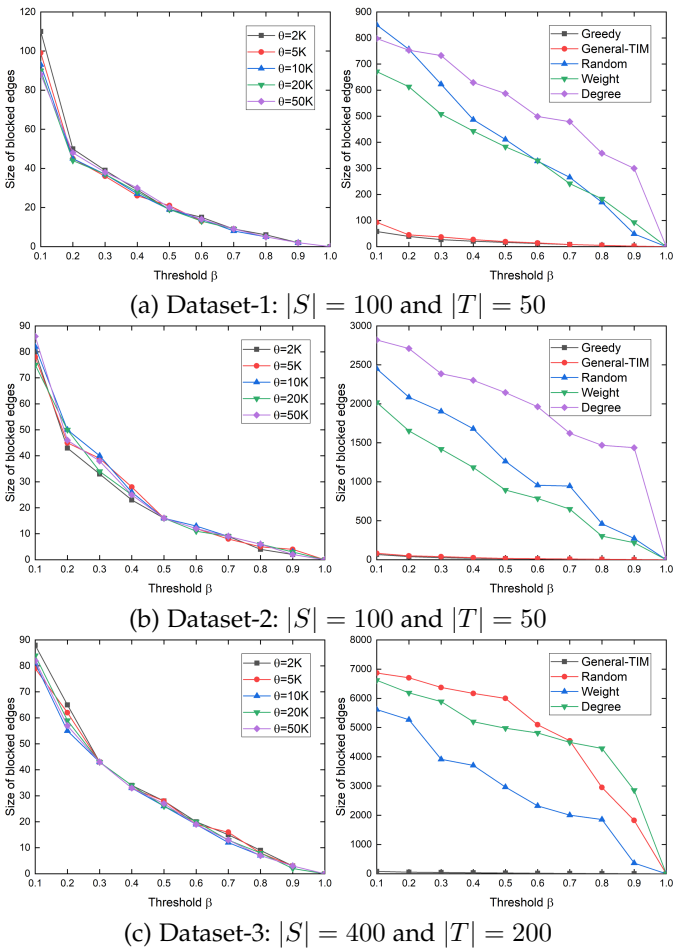


(a) Dataset-1: $|S| = 100$ and $|T| = 50$

(b) Dataset-2: $|S| = 100$ and $|T| = 50$

(c) Dataset-3: $|S| = 400$ and $|T| = 200$

Fig. 4. The performance changes over β under the Case-1. Here, the left column is achieved by General-TIM with different θ, and the right column is achieved by different algorithm.



(a) Dataset-1: $|S| = 100$ and $|T| = 50$

(b) Dataset-2: $|S| = 100$ and $|T| = 50$

(c) Dataset-3: $|S| = 400$ and $|T| = 200$

Fig. 5. The performance changes over β under the Case-2. Here, the left column is achieved by General-TIM with different θ, and the right column is achieved by different algorithm.

4.2 Experimental Setup

Two experiments are performed for each dataset. The first experiment is performed to test whether θ is large enough to get a valid solution. The second experiment is comparing our Greedy and General-TIM algorithm against some common heuristic algorithm to assess the effectiveness of the solution obtained by removing edges. The common heuristic algorithm mainly includes the following:

1) **Random:** Select the edges randomly until satisfying the ratio of β.
2) **Weight:** Select the edges from the high to low diffusion probability until satisfying the ratio of β.
3) **Degree:** Select the edges whose destination nodes with the largest number of outgoing neighbors until satisfying the ratio of β.

In the second experiment, the marginal gain for each edge is calculated by the Monte Carlo simulation, and we estimate the $f_{G(I)}(S,T)$ by simulating 1000 runs. The activation probability of each edge is uniformly distributed in $[0,1]$. The previous experiment indicates that the quality of the approximation is no significant improvement after 1000 runs. Since the computational cost of Greedy algorithm is very high, the dataset we selected cannot be too large to get solutions in a short time. Then, the value of β ranges from 0.1 to 1. Next, we need to predefine the nodes in rumor set S and targeted set T. For dataset-1 and dataset-2, we set $|S| = 100$ and $|T| = 50$, and dataset-3, $|S| = 400$ and $|T| = 200$. Then, this experiment can be divided into two cases: (1) Case-1: the nodes in S abd T are selected randamly; (2) Case-2: the nodes in S is selected with the highest outgoing degree in graph G, but nodes in T is selected randomly.

4.3 Experimental Results

Fig. 4 and Fig. 5 draw the performance achieved by different θ under General-TIM and performance comparison with other baseline algorithms under two cases and three datasets. Obviously, from the left column of Fig. 4 and Fig. 5, $\theta = 10K$ is large enough to make sure the solution is a good estimation, because the difference with $\theta = 20K$ and $\theta = 50K$ is extremely small in these three datasets. The General-TIM of right column, we assumes $\theta = 10K$.

According to what we said before, the goal of TPM problem is to select least edges to be blocked so that at most β ratio of nodes in T are influenced by rumor compared to no blocked edge. Thus, the smaller number of edges we block, the better the performance is. The right column of Fig. 4 and 5 shows the performance of the different algorithms. As depicted of that, the number of edges to be blocked

returned by Greedy algorithm is the smallest among these five algorithms, so its performance is best. Besides, the edge size of Greedy and General-TIM algorithm is almost the same, very small deviation, in two networks, which prove the effectiveness of the randomized sampling by Random-RS-Path. Comparing with Weight algorithm, which is the best heuristic algorithm, its size of blocking edges is at least 20 times than Greedy and General-TIM algorithm when $\beta = 0.1$.

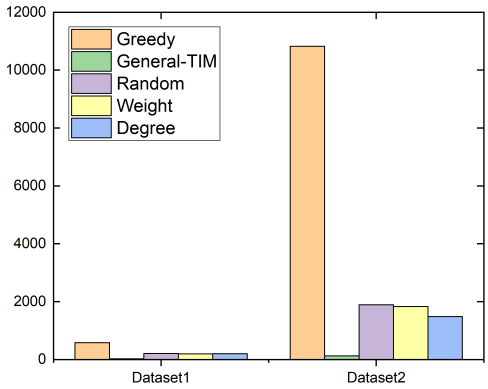


Fig. 6. The time consuming of different algorithm when $\beta = 0.1$ in the first two datasets under Case-1

Even if Greedy algorithm win with a weak gap, General-TIM is much faster than Greedy algorithm, which is shown as Fig. 6, so it is more suitable in large networks. If there is no parallel acceleration, the time consuming of Greedy algorithm is much worse than it is displayed. However, the error of edge size for General-TIM will become large when β approaching to 0. This is because, in General-TIM, we approximate that the node v is protected by blocking the shortest path from S to v, which is not unbiased estimation to protect v from influenced by rumor. For example, under Dataset-1 and Case-1, when $\beta \leq 0.01$, the relative effort between Greedy Algorithm and General-TIM will become very large, edge size in Greedy algorithm is 47, but in General-TIM is 237. Therefore, General-TIM algorithm cannot be applied to solve TPM problem when β is too small.

TABLE 2
The relative error between Greedy and General-TIM

	Dataset-1	Dataset-2	Dataset-3
Case-1	14.91%	31.22%	n/a
Case-2	17.68%	28.40%	n/a

The average relative error, threshold β from 0.1 to 1, between Greedy and General-TIM is shown as table 2. Given β, let I_1 be the edge set returned by Greedy, and I_2 returned by General-TIM, the relative error is equal to $||I_1|-|I_2||/|I_1|$. Because General-TIM is an estimation of Greedy algorithm, from table 2, we can know this estimation is valid.

5 Speedup on Community Structure

From last section, we know that General-TIM algorithm made a good result. Even if that, at line 4 of Algorithm 1 and line 6 of Algorithm 4, we need to check each edge so as to get the edge with maximum marginal gain.

5.1 Speedup Strategy

An apparent method to reduce the computational cost of selecting edge with maximum marginal gain is to reduce the size of candidate edges. In order to achieve this goal, we notice that a phenomenon that is common in the social network: community structure. Communities are groups of vertices which have the same attributes and play a similar role in the networks. For example, in online social network (OSN), which is established from friendship relation, such as Facebook or Linkedln, all people with the same interests, or hobbies, in activities, who tend to communicate each other frequently may form a community. In a community, the influence can be spread faster than outside. Therefore, if we know the community structure of network, we have a deep understanding about the organization of relation structures. In graph $G = (V, E)$, a community structure $\mathcal{C}$, where $\mathcal{C} = \{C_1, C_2, ..., C_n\}$, is a partition of V, which means that for any $i, j \in \{1, 2, ..., n\}$, $C_i \cap C_j = \emptyset$. In order to reduce the size of candidate edges, we consider an instance of TPM problem. Here, the nodes in rumor set S and targeted set T do not appear in some communities, then we do not need to consider the edges in these communities when selecting an edge to be blocked in General-TIM or Greedy Algorithm.

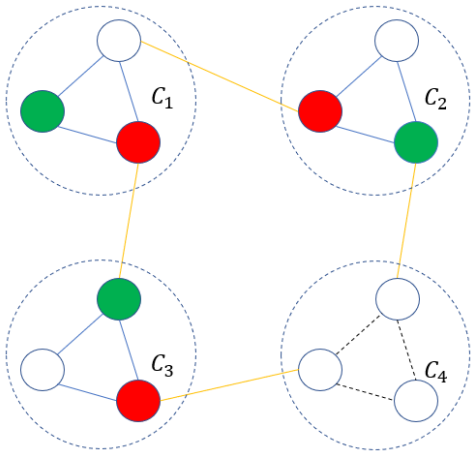


Fig. 7. An example to show how speedup strategy based on community structure works

For example, from Fig. 7, there are four communities $\mathcal{C} = \{C_1, C_2, C_3, C_4\}$. We know that, in community C_4, there is no node in rumor set S and targeted set T, thus, we do not need to consider edge (black dotted line edges) in community C_4 when selecting edge with maximum marginal gain into the solution, thereby reduce the size of candidate edges. The above example shows the huge benefit of exploit community structure, we propose such a speedup strategy based on community structure, which can reduce the computational cost when some communities with no node in rumor set S and targeted set T exist. Given $G = (V, E), S, T$ and β, the process of speedup strategy can be described as follows:

1) Use some existing algorithm to obtain the community structure of G

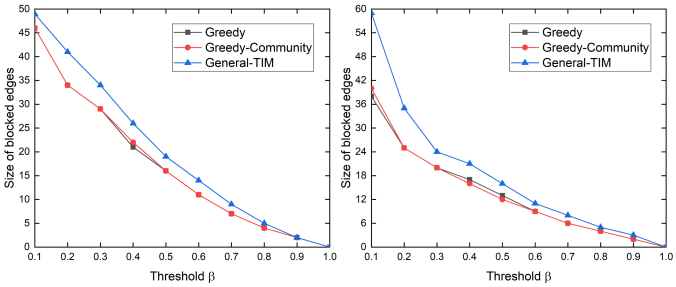


Fig. 8. The performance changes over β under the Case-1 with speedup strategy. Here, the left column is under the Dataset-1, and the right column is under the Dataset-2.

2) Remove the edges in community with no node in S and T from E, we can get a new candidate edge set E'. Obviously, $|E'| \leq |E|$.
3) Select an edge $e \in E' \backslash I$ to obtain maximum marginal gain under the Algorithm 1 or Algorithm 4 iteratively under satisfying β.
4) Return edge set I that should be blocked.

It is worth noting that we cannot delete the bridge edges (yellow edges in Fig. 7), because rumor node in C_3 may influences targeted node in C_2 through the edges in C_4. Given community structure, the influence spread within the same community is of high probabilities, and the influence spread across communities is of low probabilities. However, it cannot be avoided the influence spread across communities. Thus, the bridge must be considered even if they may be not the optimal solution. As we are known, network communities are sets of nodes with lots of internal connections and few external ones to the rest of the network. In above process, we need to find the community structure for the given network, and there are many different methods to find the community structure. In this paper, we use the classical Clauset-Newman-Moore greedy modularity maximization. Greedy modularity maximization begins with each node in its own community and joins the pair of communities that most increases modularity until no such pair exists.

5.2 Experimental Results

The experimental setup is the same as the former experiment. The number of communities found by Clauset-Newman-Moore greedy modularity maximization are 19 in Dataset-1 and 13 in Dataset-2 respectively. In Dataset-1, the nodes in rumor set and targeted set are distributed in half of the communities, and in Dataset-2, the nodes in that are distributed in one-third of the communities. The edges in those communities with no rumor and targeted nodes will be removed from candidate edge set E'. In order to show the difference clearly, other baseline algorithms will be removed in the following Fig. 8.

Fig. 8 draws the performance comparison achieved by Greedy and Greedy based on community structure with different threshold β. We see that the edge size selected from Greedy Algorithm based on community structure is very close to that from Greedy Algorithm, but the computational cost is reduced significantly. The reduced time is proportional to the edge size that is removed. Similarly, General-TIM can achieve the same results with the help of speedup strategy.

6 Conclusion

In this paper, we modeled the problem of rumor blocking to targeted set in social networks. Targeted Protection Maximization (TPM) was formulated and we proved that it is monotone non-decreasing, but not submodular and supermodular. We proved an upper bound to the Greedy solution. Then, we proposed the General-TIM algorithm with the help of Random-RS-Path. Then, we represented a speedup strategy based on community structure to speed up both Greedy and General-TIM algorithm. Finally, we tested our algorithms on three real-world datasets. The experimental result verified the effectiveness and efficiency of General-TIM algorithm and speedup strategy.

Acknowledgments

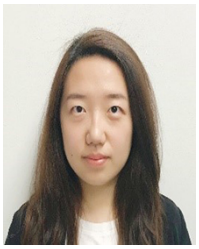
This work is partly supported by National Science Foundation under grant 1747818.

References

[1] H. Allcott and M. Gentzkow, "Social media and fake news in the 2016 election," *Journal of economic perspectives*, vol. 31, no. 2, pp. 211–36, 2017.
[2] D. Kempe, J. Kleinberg, and É. 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*. ACM, 2003, pp. 137–146.
[3] C. Budak, D. Agrawal, and A. El Abbadi, "Limiting the spread of misinformation in social networks," in *Proceedings of the 20th international conference on World wide web*. ACM, 2011, pp. 665–674.
[4] L. Fan, Z. Lu, W. Wu, B. Thuraisingham, H. Ma, and Y. Bi, "Least cost rumor blocking in social networks," in *2013 IEEE 33rd International Conference on Distributed Computing Systems*. IEEE, 2013, pp. 540–549.
[5] L.-l. Ma, C. Ma, H.-F. Zhang, and B.-H. Wang, "Identifying influential spreaders in complex networks based on gravity formula," *Physica A: Statistical Mechanics and its Applications*, vol. 451, pp. 205–212, 2016.
[6] S. Wang, X. Zhao, Y. Chen, Z. Li, K. Zhang, and J. Xia, "Negative influence minimizing by blocking nodes in social networks," in *Workshops at the Twenty-Seventh AAAI Conference on Artificial Intelligence*, 2013.
[7] E. B. Khalil, B. Dilkina, and L. Song, "Scalable diffusion-aware optimization of network topology," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014, pp. 1226–1235.
[8] M. Kimura, K. Saito, and H. Motoda, "Minimizing the spread of contamination by blocking links in a network." in *AAAI*, vol. 8, 2008, pp. 1175–1180.
[9] H. Tong, B. A. Prakash, T. Eliassi-Rad, M. Faloutsos, and C. Faloutsos, "Gelling, and melting, large graphs by edge manipulation," in *Proceedings of the 21st ACM international conference on Information and knowledge management*. ACM, 2012, pp. 245–254.
[10] G. Tong, W. Wu, L. Guo, D. Li, C. Liu, B. Liu, and D.-Z. Du, "An efficient randomized algorithm for rumor blocking in online social networks," *IEEE Transactions on Network Science and Engineering*, 2017.
[11] P. Wu and L. Pan, "Scalable influence blocking maximization in social networks under competitive independent cascade models," *Computer Networks*, vol. 123, pp. 38–50, 2017.
[12] N. Arazkhani, M. R. Meybodi, and A. Rezvanian, "An efficient algorithm for influence blocking maximization based on community detection," in *2019 5th International Conference on Web Research (ICWR)*. IEEE, 2019, pp. 258–263.

[13] W. Chen, C. Wang, and 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*. ACM, 2010, pp. 1029–1038.
[14] C. Borgs, M. Brautbar, J. Chayes, and B. Lucier, "Maximizing social influence in nearly optimal time," in *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*. SIAM, 2014, pp. 946–957.
[15] Y. Tang, X. Xiao, and Y. Shi, "Influence maximization: Near-optimal time complexity meets practical efficiency," in *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*. ACM, 2014, pp. 75–86.
[16] Y. Tang, Y. Shi, and X. Xiao, "Influence maximization in near-linear time: A martingale approach," in *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*. ACM, 2015, pp. 1539–1554.
[17] X. He, G. Song, W. Chen, and Q. Jiang, "Influence blocking maximization in social networks under the competitive linear threshold model," in *Proceedings of the 2012 siam international conference on data mining*. SIAM, 2012, pp. 463–474.
[18] N. P. Nguyen, G. Yan, M. T. Thai, and S. Eidenbenz, "Containment of misinformation spread in online social networks," in *Proceedings of the 4th Annual ACM Web Science Conference*. ACM, 2012, pp. 213–222.
[19] S. Kumar and N. Shah, "False information on web and social media: A survey," *arXiv preprint arXiv:1804.08559*, 2018.
[20] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance, "Cost-effective outbreak detection in networks," in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2007, pp. 420–429.
[21] A. Goyal, W. Lu, and L. V. Lakshmanan, "Celf++: optimizing the greedy algorithm for influence maximization in social networks," in *Proceedings of the 20th international conference companion on World wide web*. ACM, 2011, pp. 47–48.
[22] J. Guo and W. Wu, "A novel scene of viral marketing for complementary products," *IEEE Transactions on Computational Social Systems*, vol. 6, no. 4, pp. 797–808, Aug 2019.
[23] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functionsi," *Mathematical programming*, vol. 14, no. 1, pp. 265–294, 1978.
[24] E. Khalil, B. Dilkina, and L. Song, "Cuttingedge: influence minimization in networks," in *Proceedings of Workshop on Frontiers of Network Analysis: Methods, Models, and Applications at NIPS*, 2013.
[25] C. V. Pham, Q. V. Phu, and H. X. Hoang, "Targeted misinformation blocking on online social networks," in *Asian Conference on Intelligent Information and Database Systems*. Springer, 2018, pp. 107–116.
[26] M. Conforti and G. Cornuéjols, "Submodular set functions, matroids and the greedy algorithm: tight worst-case bounds and some generalizations of the rado-edmonds theorem," *Discrete applied mathematics*, vol. 7, no. 3, pp. 251–274, 1984.
[27] J. Vondrák, "Submodularity and curvature: The optimal algorithm (combinatorial optimization and discrete algorithms)," 2010.
[28] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.
[29] A. Das and D. Kempe, "Submodular meets spectral: Greedy algorithms for subset selection, sparse approximation and dictionary selection," *arXiv preprint arXiv:1102.3975*, 2011.
[30] A. A. Bian, J. M. Buhmann, A. Krause, and S. Tschiatschek, "Guarantees for greedy maximization of non-submodular functions with applications," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 498–507.
[31] R. A. Rossi and N. K. Ahmed, "The network data repository with interactive graph analytics and visualization," in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015. [Online]. Available: http://networkrepository.com

Jianxiong Guo is a Ph.D candidate in the Department of Computer Science at the University of Texas at Dallas. He received his BS degree in Energy Engineering and Automation from South China University of Technology in 2015 and MS degree in Chemical Engineering from University of Pittsburgh in 2016. His research interests include social networks and design of approximation algorithm.



Yi Li received her MS degree in Computer Science and Digital Communication/Multimedia from University of Texas at Dallas. She is a PhD candidate in the Department of Computer Science in University of Texas at Dallas. Her research area include social network analysis and algorithm design.

Weili Wu received the Ph.D. and M.S. degrees from the Department of Computer Science, University of Minnesota, Minneapolis, MN, USA, in 2002 and 1998, respectively. She is currently a Full Professor with the Department of Computer Science, The University of Texas at Dallas, Richardson, TX, USA. Her research mainly deals in the general research area of data communication and data management. Her research focuses on the design and analysis of algorithms for optimization problems that occur in wireless networking environments and various database systems.