This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2019.2903272, IEEE Transactions on Network Science and Engineering

Minimizing Influence of Rumors by Blockers on Social Networks: Algorithms and Analysis

Ruidong Yan, Deying Li, Weili Wu, Ding-Zhu Du and Yongcai Wang

Abstract—Online social networks such as Facebook, Twitter and Wechat have become major social tools. The users can not only keep in touch with family and friends, but also send and share the instant information. However, in some practical scenarios, we need to take effective measures to control the negative information spreading, e.g, rumors spread over the networks. In this paper, we first propose the *Minimizing Influence of Rumors* (MIR) problem, i.e., selecting a blocker set $\mathcal{B}$ with k nodes such that the users' total activation probability by rumor source set S is minimized. Then we employ the classical *Independent Cascade* (IC) model as information diffusion model. Based on the IC model, we prove the objective function is monotone decreasing and non-submodular. To address the MIR problem effectively, we propose a two-stages method *Generating Candidate Set* & *Selecting Blockers* (GCSSB) for the general networks. Furthermore, we also study the MIR problem on the tree network and propose a dynamic programming guaranteeing the optimal solution. Finally, we evaluate proposed algorithms by simulations on synthetic and real-life social networks, respectively. Experimental results show our algorithms are superior to the comparative heuristic approaches such as *Out-Degree* (OD), *Betweenness Centrality* (BC) and *PageRank* (PR).

Index Terms—Social network, rumor blocking, submodularity, greedy algorithm, dynamic programming.

1 Introduction

With the advance of the internet and computer technology, some significant social networks have been widely integrated into our daily life such as Facebook, Twitter, Google+ and Wechat. Social networks can usually be represented as complex networks of nodes and edges, where nodes denote the users (people, organizations, or other social entities) and edges denote the social relationships between users (friendship, collaboration, or information interaction). The users in these online social networks can not only disseminate the positive contents (ideas, opinions, innovations, interests and so on) but also the negative information such as rumors. It has been shown that rumors spread very fast and cause serious consequences [1]. For example, when the devastating wild-fires happens in California in October 2017, at the time the officers were evacuating residents and searching through the burned ruins of homes for missing persons they still had to deal with the fake news. Although the rumor was shot down by the officers and was debunked by some government websites afterwards, the original story was shared 60,000 times and similar stories was shared 75,000 times on Facebook.

- R. Yan is with the School of Information, Renmin University of China, Beijing, 100872, China. E-mail: yanruidong@ruc.edu.cn
- D. Li is with the School of Information, Renmin University of China, Beijing, 100872, China E-mail: deyingli@ruc.edu.cn (Corresponding author)
- W. Wu is with the Department of Computer Science, University of Texas at Dallas, TX 75080, USA E-mail: weiliwu@utdallas.edu
- D. Du is with the Department of Computer Science, University of Texas at Dallas, TX 75080, USA E-mail: dzdu@utdallas.edu
- Y. Wang is with the School of Information, Renmin University of China, Beijing, 100872, China E-mail: ycw@ruc.edu.cn

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

In order to provide quality service and accurate information, it is crucial to have an effective strategy to block or limit the negative effects of such rumors. Limiting the rumors spreading in social networks is a hot but challenging research topic [2]. Currently, the literature related to rumor blocking can be roughly divided into the following three categories: (1) Block rumors by the influential nodes. This method usually selects the most influential nodes in the networks based on certain criteria and removes these influential nodes from original network such that the rumors spreading is limited such as [3], [4], [5], [6], [7]; (2) Block rumors by the key edges. This method usually removes a set of edges that play a key role in information dissemination such that the rumor spreading is as less as possible [8], [9], [10], [11], [12]; (3) Spread the positive information (e.g., truth) to clarify the rumors. This method is based on the assumption that users will not adopt the rumors once they have adopted the positive truth. More specifically, it identifies a subset of nodes and disseminates the positive information such that the positive information is adopted by as many users as possible [13], [14], [15], [16].

In this paper, we study a novel *Minimizing Influence of Rumors* (MIR) problem to limit the rumors spreading, i.e., we want to identify a set $\mathcal{B}$ with k nodes and remove this set from the original network such that the total activation probability of nodes by the rumor source set S on modified network is minimized. We call a node $v \in \mathcal{B}$ a *blocker*. To solve the MIR problem effectively, we propose a two-stages method GCSSB that includes *Generating Candidate Set* (GCS) and *Selecting Blockers* (SB) stages. Specifically, in the GCS stage, we sort the nodes on the network to find the top $\alpha \cdot k$ nodes that have strong ability to disseminate the rumors, where α is a threshold parameter. And we generate a candidate set of blockers $\mathcal{C}$ by these $\alpha \cdot k$ nodes. In the SB stage, we design a basic greedy algorithm and select k

nodes from the previous candidate set $\mathcal{C}$ according to the maximum marginal gain. Different the previous researches, we have a preprocessing stage before we design the basic greedy algorithm. The advantage is that it can effectively reduce the time consumption of the greedy algorithm. In other words, we identify the blocker set from the subgraph based on the candidate set $\mathcal{C}$ instead of the original network. In addition, we also explore the MIR problem in a special structure such as a tree network.

We summarize our main contributions as follows:

- We formalize the *Minimizing Influence of Rumors* (MIR) problem and prove the objective function is not submodular under the *Independent Cascade* (IC) model.
- We propose a two-stages strategy named GCSSB to solve the MIR problem on general social networks for the first time.
- We also study the MIR problem on a special network such as a tree, and we provide a dynamic programming algorithm to guarantee the optimal solution.
- In order to evaluate proposed algorithms, we use a synthetic and three real-life social networks with various scales in experiments. Furthermore, we compare proposed method with other heuristic approaches. Experimental results validate that our methods are superior to other approaches.

The rest of this paper is organized as follows. We first begin by recalling some existing related work of the rumor blocking in Section 2. Then we introduce the information diffusion model in Section 3. And we show the problem description and properties in Section 4. Algorithm for the general networks is presented in Section 5. The dynamic programming algorithm on the tree network is proposed in section 6. We analyze and discuss the results of the experiments in Section 7. Finally, we draw our conclusions in Section 8.

2 RELATED WORK

Domingos et al. [17] first study the influence between users for marketing in social networks. Kempe et al. [18] model the viral marketing as a discrete optimization problem, which is named *Influence Maximization* (IM). They propose a greedy algorithm with $(1-1/e)$-approximation ratio since the function is submodular under *Independent Cascade* (IC) or *Linear Threshold* (LT) model. Based on Kempe's contributions, there have been substantial efforts in modeling the information propagation in recent years such as [19], [20], [21].

2.1 Blocking Nodes for Rumors

In [3], Fan et al. study a problem that identifies a minimal subset of individuals as initial protectors (The nodes are used to limit the bad influence of rumors.) to minimize the number of people infected in neighbor communities at the end of both diffusion processes. Authors propose algorithms under the *Opportunistic One-Activate-One* as well as the *Deterministic One-Activate-Many* models, and they show the theoretical analysis in detail. In [4], Wang et al. address the problem of minimizing the influence of rumors. They assume a rumor emerges and affects some users in the social network. And they want to minimize the size of ultimately contaminated users by discovering k uninfected users. A simple greedy method is proposed. Unfortunately, they have no theoretical analysis. In social networks, how to identify the influential spreaders is crucial for rumors. Ma et al. in [5] propose a gravity centrality index to identify the influential spreaders in complex networks, and they compare with some well-known centralities such as degree, betweenness, closeness and so forth.

2.2 Blocking Links for Rumors

In [8], Kimura et al. propose a method (by blocking a limited number of links) for efficiently finding a good approximate solution to rumor blocking. In [9], Khalil et al. abstract the flu control problem into an edge deletion problem. They show this problem is supermodular under the LT model. Based on this property, they design a scalable algorithm with approximation guarantees. In [10], Tong et al. propose effective and scalable algorithms to solve dissemination problems and answer which edges should be deleted in order to contain a rumor.

2.3 Spreading Positive Truth for Rumors

In [13], Budak et al. study the two competing campaigns (rumor and truth) simultaneously spreading in a network. They address the problem of limiting the rumor propagating. In other words, they want to identify a subset of individuals to spread the truth such that as many nodes as possible in the network adopt the truth rather than the rumor at the end of both propagation processes. And they show this problem is NP-hard and provide a greedy algorithm. In [14], Tong et al. study the rumor blocking problem that asks for k seed users to trigger the spread of a positive cascade such that the number of the users who are not influenced by rumor can be maximized. They present a randomized approximation algorithm which is provably superior to the state-of-the art methods with respect to running time.

2.4 Studies in the Tree Network

The social influence problems based on tree networks have also attracted attention such as [22], [23]. In [22], Lappas et al. consider the problem of selecting a set of k active nodes to explain the observed activation state under a given information-propagation model. And they show that, in trees, this problem can be solved in polynomial time by a dynamic programming. Bharathi et al. in [23] study the game of multiple competing innovation diffusions when multiple companies market competing products using the viral marketing. And they give a FPTAS for the problem of maximizing the influence of a single player when the underlying graph is a tree.

In this paper, we study MIR problems on general networks (non-tree networks) and tree networks, respectively. We find that the MIR problem is difficult to find the optimal solution on the general network and proposes a two-stage algorithm. On the other hand, we show that the MIR problem can find the optimal solution on the tree network and propose a dynamic programming algorithm.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2019.2903272, IEEE Transactions on Network Science and Engineering

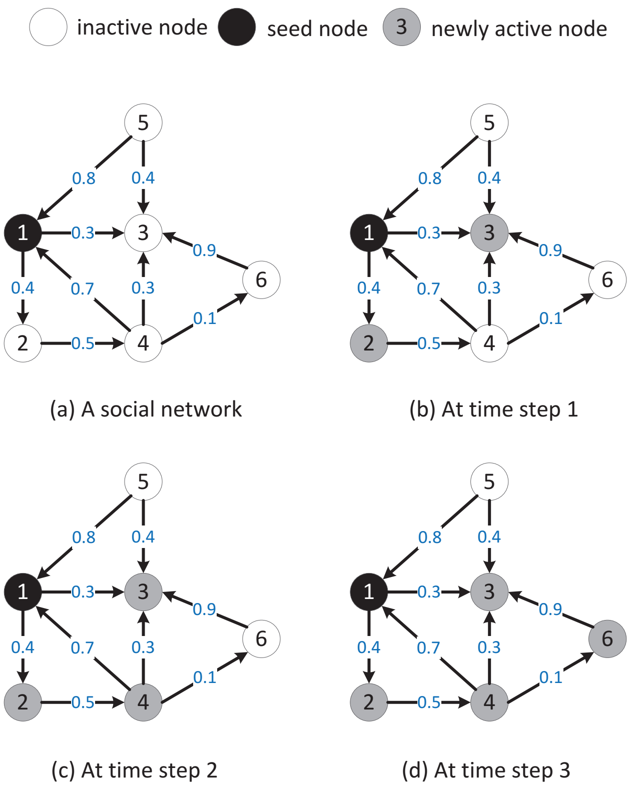


Fig. 1. An example to illustrate the independent cascade model.

3 Information Diffusion Model

In this section, we briefly introduce influence diffusion model: *Independent Cascade* (IC) model which is first proposed by [18]. Given a directed social network that can be denoted by a graph $G = (V, E, p)$, V represents users (node set), $E \subseteq V \times V$ represents the relationships between users (edge set), and p_{uv} of the edge (u, v) denotes the probability that the node u activates v. We call a node *active* if it adopts the information (rumor) from other nodes, *inactive* otherwise. Influence propagation process unfolds discrete time steps t_i, where $i = 0, 1, ..., n$ and n is number of nodes in G.

More specifically, let S_{t_0} be the initial source nodes of rumor, i.e., seed set. Let S_{t_i} denote *active* nodes in time step t_i, and each node u in S_{t_i} has single chance to activate each *inactive* neighbor v through its out-edge (u, v) with the probability p_{uv} at time step t_{i+1}. Repeat this process until no more new nodes can be activated. Note that a node can only switch from *inactive* to *active*, but not in the reverse direction.

In Fig. 1, there is an example to illustrate the independent cascade model in detail. The nodes in figure have three states: *inactive*, *active* and *newly active*. The number embeded on each edge (u, v) denotes the propagation probability, e.g., the 0.1 on the edge $(4, 6)$ denotes the probability that the node 4 activates the node 6. In particular, only the seed nodes are active and other nodes are inactive at time step 0. Fig. 1(a) shows a simple social network with the seed node 1 at time step 0. And the active seed node 1 attempts to activate its inactive neighbors (node 2 and node 3) at this time. At time step 1, we can observe that the node 2 and the node 3 become newly active[1] in Fig. 1(b). And at this time, the newly active node 2 and node 3 activate their own inactive neighbors, respectively. Then the node 4 becomes active in Fig. 1(c). Analogously, at time step 2, the newly active node 4 attempts to activate its inactive neighbor, i.e., node 6. Finally, the node 6 becomes active at time step 3. In addition, the information dissemination process stops since no more nodes can be activated.

4 Problem Description and Properties

4.1 Problem Description

Given a directed social network $G = (V, E, p)$, an information diffusion model $\mathcal{M}$ and a rumor source set S, V denotes user set, $E \subseteq V \times V$ denotes the relationships between users, and p_{uv} of edge (u, v) denotes the probability that u activates v successfully. We define the activation probability of a node $v \in V$ by seed set S under model $\mathcal{M}$ as following equation.

$$Pr_{\mathcal{M}}(v, S) = \begin{cases} 1, & \text{if } v \in S \\ 0, & \text{if } N^{in}(v) = \emptyset \\ 1 - \prod\limits_{u \in N^{in}(v)} (1 - Pr_{\mathcal{M}}(u, S) p_{uv}), & \text{otherwise.} \end{cases} \tag{1}$$

Where $N^{in}(v)$ is the set of in-neighbors of v and $Pr_{\mathcal{M}}(u, S)p_{uv}$ represents the probability u successfully activates v under the diffusion model $\mathcal{M}$ (Here, $\mathcal{M}$ is the IC model). We can clearly see that the activation probability of a node v depends on the its in-neighbors u. Here, we take the node 1 activating the node 4 as an example to illustrate how to calculate the activation probability in Fig. 1, i.e., $Pr_{\mathcal{M}}(4, \{1\})$.

According to the equation (1), we have

$$Pr_{\mathcal{M}}(4, \{1\}) = 1 - \prod_{u \in N^{in}(4)} (1 - Pr_{\mathcal{M}}(u, \{1\}) p_{u4}), \tag{2}$$

where $N^{in}(4) = \{2\}$ and the equation (2) can be converted to the following equation

$$Pr_{\mathcal{M}}(4, \{1\}) = 1 - (1 - Pr_{\mathcal{M}}(2, \{1\}) p_{24}). \tag{3}$$

Similarly,

$$Pr_{\mathcal{M}}(2, \{1\}) = 1 - (1 - Pr_{\mathcal{M}}(1, \{1\}) p_{12}),$$

where $Pr_{\mathcal{M}}(1, \{1\}) = 1$ since the node 1 is the seed node and $p_{12} = 0.4$. Therefore, $Pr_{\mathcal{M}}(2, \{1\}) = 0.4$. Add $Pr_{\mathcal{M}}(2, \{1\}) = 0.4$ into the equation (3), we have $Pr_{\mathcal{M}}(4, \{1\}) = 1 - (1 - 0.4 \times 0.5) = 0.2$.

Now we give the problem description as follow.

Definition 1. Minimizing Influence of Rumor (MIR). Given a directed social network $G = (V, E, p)$, a rumor source set S, a positive integer budget k, and the IC model $\mathcal{M}$, MIR aims to find a blocker set $\mathcal{B}$ with k nodes such that

$$\mathcal{B}^* = \arg \min_{\mathcal{B} \subseteq V \setminus S, |\mathcal{B}| = k} \sum_{v \in V \setminus \{S \cup \mathcal{B}\}} Pr_{\mathcal{M}}(v, S). \tag{4}$$

1. In fact, from the previous description, we can easily find that the independent cascade model is a random process. That is to say, the active nodes activating the inactive nodes is random. Here, we only show one of the possible scenarios.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2019.2903272, IEEE Transactions on Network Science and Engineering

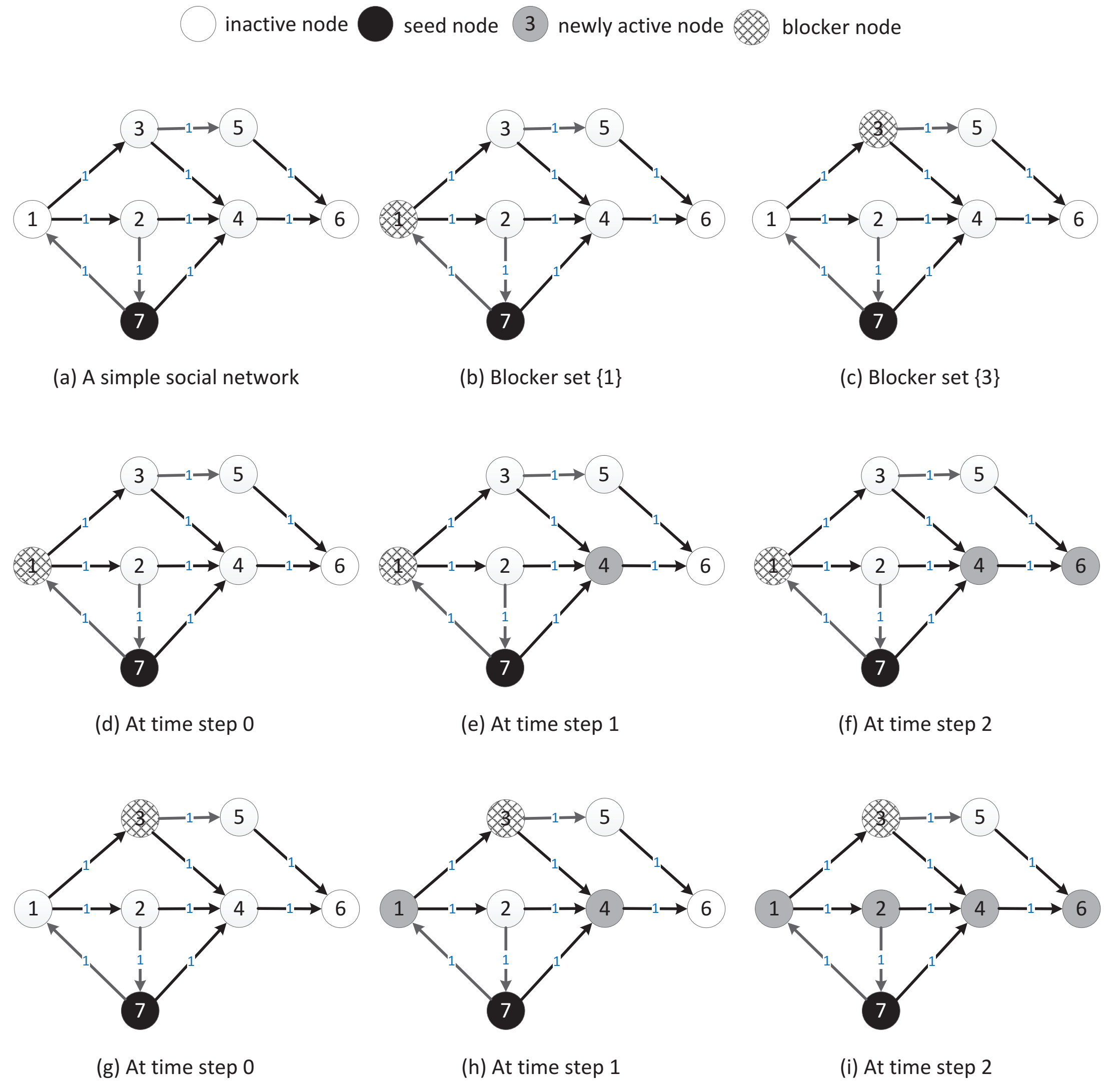


Fig. 2. An example shows calculating the value of objective function when $k = 1$. Fig. 2(a) shows a simple social network where the black solid node indicates the rumor source $S = \{7\}$. Fig. 2(b) shows selecting the node 1 as a blocker. Fig. 2(c) shows selecting the node 3 as a blocker.

Fig. 2 shows how to calculate the objective function value if we select only one node ($k = 1$) as a blocker under the independent cascade model. For ease of exposition, we let the node 7 be the rumor seed and the propagation probability of each edge in the network is 1[2]. More specifically, Fig. 2(a) shows a simple social network. We calculate the following two cases: (1) Selecting the node 1 as the blocker set $\mathcal{B}$ (see Fig. 2(b)); (2) Selecting the node 3 as the blocker set $\mathcal{B}$ (see Fig. 2(c)).

In the first case, Fig. 2(d)-Fig. 2(f) show the information dissemination process when we select the node 1 as a blocker. Note that once a node becomes a blocker it will not be activated. At time step 0, the node 7 attempts to activate its inactive neighbor. At time step 1, the node 4 becomes active since the propagation probability is 1. And the node 4 is ready to activate its inactive neighbor. At time step 2, the node 6 becomes active and information dissemination process stops. As a result, we observe that two nodes (node 4 and node 6) become newly active nodes except the seed node. Therefore the total activation probability $\sum_{v \in V \setminus \{S \cup \mathcal{B}\}} Pr_{\mathcal{M}}(v, S) = 2$.

In the second case, Fig. 2(g)-Fig. 2(i) show the information dissemination process when we select node 3 as a blocker. At time step 0, the node 7 attempts to activate its inactive neighbor. At time step 1, the node 1 and the node 4 become active since the propagation probabilities are 1. And the node 1 and node 4 attempt to activate their inactive neighbors. At time step 2, the node 2 and node 6 become active and information dissemination process stops. Consequently, there are four nodes becoming active (node 1, node 2, node 4 and node 6). Therefore the total activation probability $\sum_{v \in V \setminus \{S \cup \mathcal{B}\}} Pr_{\mathcal{M}}(v, S) = 4$. In summary, it is more appropriate to select node 1 as a blocker instead of node 3 for the MIR problem.

2. In this case, the information dissemination process is determined.

4.2 Properties of Objective Function

Theorem 1. The objective function (4) is monotone decreasing and not submodular under the IC model.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2019.2903272, IEEE Transactions on Network Science and Engineering

Proof 1. A set function f is monotone decreasing if $f(A) \geq f(B)$ whenever $A \subseteq B$. It is obvious that the objective function is monotone decreasing in our MIR problem, because the more blockers we choose, the smaller the objective function will be. We omit its proof.

Then we show the objective function is not submodular. If V is a finite set, a submodular function is a set function $f : 2^V \rightarrow \Re$, where 2^V denotes the power set of V, satisfies the following condition: for every $A \subseteq B \subseteq V$ and $x \in V \backslash B$, $f(A \cup \{x\}) - f(A) \geq f(B \cup \{x\}) - f(B)$. Going back to the example mentioned in Fig. 2, we let rumor seed $S = \{7\}$, $f = \sum_{v \in V \backslash \{S \cup \mathcal{B}\}} Pr_{\mathcal{M}}(v, S)$, $A = \{4\}$, $B = \{3, 4\}$ and $x = \{1\}$. The $f(A)$ means that we select the node 4 as the blocker set $\mathcal{B}$. In this case, we analyze the information dissemination process. At time step 0, the node 7 activates the node 1. At time step 1, the node 1 becomes active and attempts to activate its inactive neighbors. At time step 2, the node 2 and the node 3 become active and they attempt to activate their inactive neighbors. At time step 3, the node 5 becomes active and attempts to activate its inactive neighbor. At time step 4, the node 6 becomes active and the information dissemination process stops. Finally, we can see that there are five nodes become active, i.e., $f(A) = 5$. Similarly, we compute the $f(A \cup \{x\}) = 0$, $f(B) = 2$ and $f(B \cup \{x\}) = 0$. Therefore $f(A \cup \{x\}) - f(A) < f(B \cup \{x\}) - f(B)$, which indicates non-submodularity of MIR.

5 Algorithm for The General Network

In this section, we propose a two-stages method GCSSB which includes generating candidate set and selecting blockers stages. We introduce them in Section 5.1 and Section 5.2, respectively.

5.1 Generating Candidate Set

Given a directed social network $G = (V, E, p)$ and a rumor source set S, we first sort nodes on the network. The purpose of sorting is to determine the candidate set of blockers and reduce time consuming by the greedy algorithm in second stage. Intuitively, we will choose the nodes with strong spreading ability as blockers rather than those nodes with weak spreading ability. Therefore, how to measure the spreading ability of nodes becomes a key issue.

Here, we define a vector $\sigma = I + AI + ...A^r I$, where A denotes the adjacent matrix of network and I denotes unit column vector and $1 \leq r \leq |V|$. As we all know, A^r_{ij} denotes the approximation probability that i activates j through a path of length r. Therefore σ denotes its total probability. For example, Fig. 3(a) shows the adjacent matrix of the network in Fig. 2 when the propagation probability $p = 0.5$ for each edge in network. Fig. 3(b) shows $\sigma = I + AI + ...A^r I = (3.938, 3.531, 2.5, 1.5, 1.5, 1, 3.656)^T$ where $r = 5$. And we sort σ in descending order. Then we obtain permutation $\Pi = (3.938, 3.656, 3.531, 2.5, 1.5, 1.5, 1)^T$ and choose the top $\alpha * k$ nodes as the candidate set of blockers $\mathcal{C}$ where α is a threshold parameter (In the experiment section, we set the parameter α from 1 to 10). Consistent with the example mentioned earlier, we should choose node 1 as a blocker instead of the node 3 because $\sigma_1 = 3.938 > \sigma_3 = 2.5$ when the budget $k = 1$.

5.2 Selecting Blockers

In subsection 5.1, we first determine the candidate set of the blockers $\mathcal{C}$. And in this subsection, we introduce how to accurately select k blockers based on the maximum marginal gain from the candidate set $\mathcal{C}$. Specifically, we propose a greedy algorithm based on maximum marginal gain. We give the definition of marginal gain as follow.

Definition 2. (Marginal Gain). Given a directed social network $G = (V, E, p)$, a rumor source set S and information diffusion model $\mathcal{M}$, for any node $x \in V \backslash S$, let

$$\Delta(x|S) = \sum_{v \in V \backslash S} Pr_{\mathcal{M}}(v, S) - \sum_{v \in V \backslash \{S \cup \{x\}\}} Pr_{\mathcal{M}}(v, S)$$

be marginal gain of S with respect to x.

Obviously, our algorithm focuses on the maximum marginal gain of nodes in candidate $\mathcal{C}$. We define following

Definition 3. (Maximum Marginal Gain). For any node $x \in \mathcal{C}$, let

$$x^* = \arg\max_{x \in \mathcal{C}} \Delta(x|S) = \arg\max_{x \in \mathcal{C}} \Big(\sum_{v \in V \backslash S} Pr_{\mathcal{M}}(v, S) - \sum_{v \in V \backslash \{S \cup \{x\}\}} Pr_{\mathcal{M}}(v, S) \Big)$$

be the maximum marginal gain of S with respect to x.

Algorithm 1 . Greedy Algorithm (GA)

Input: $G = (V, E, p)$, $\mathcal{M}$, S, $\mathcal{C}$ and k.
Output: $\mathcal{B}$.

1: $\mathcal{B}_0 \leftarrow \emptyset$, $\Delta(x|S) = 0$ for $x \in \mathcal{C}$;
2: **for** $t = 1$ to k **do**
3:   **for** each $x \in \mathcal{C}$ **do**
4:     $\Delta(x|S) = \sum_{v \in V \backslash \{S \cup \mathcal{B}_t\}} Pr_{\mathcal{M}}(v, S) - \sum_{v \in V \backslash \{S \cup \mathcal{B}_t \cup \{x\}\}} Pr_{\mathcal{M}}(v, S)$;
5:   **end for**
6:   $x^* = \arg\max_{x \in \mathcal{C}} \Delta(x|S)$;
7:   $\mathcal{B}_t \leftarrow \mathcal{B}_t \cup \{x^*\}$;
8:   $\mathcal{C} \leftarrow \mathcal{C} \backslash \{x^*\}$;
9: **end for**
10: **return** $\mathcal{B} \leftarrow \mathcal{B}_t$.

Based on the above definitions, we propose our greedy algorithm. We first start with the empty set, i.e., $\mathcal{B} = \emptyset$. Then, in the t-th iteration, we add the node x_t who has the maximum marginal gain into the current $\mathcal{B}$. The algorithm executes k times until k blockers are selected. The greedy algorithm is shown in Algorithm 1.

Let us analyze the complexity of the Algorithm 1. The loop from line 2 to 9 at most runs k times. In each iteration, the inner loop runs at most $|\mathcal{C}|$ times and it takes at most $O(|E|)$ time to calculate $\Delta(x|S)$. Therefore, the total time complexity is $O(k|\mathcal{C}||E|)$ in the worst case.

$$A = \begin{bmatrix} 0 & 0.5 & 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.5 & 0 & 0 & 0.5 \\ 0 & 0 & 0 & 0.5 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.5 & 0 & 0 & 0.5 & 0 & 0 & 0 \end{bmatrix}$$

(a) Adjacent matrix

$$I = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \quad AI = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0.5 \\ 0.5 \\ 0 \\ 1 \end{bmatrix}, \quad A^2 I = \begin{bmatrix} 1 \\ 0.75 \\ 0.5 \\ 0 \\ 0 \\ 0 \\ 0.75 \end{bmatrix} \dots \ A^5 I = \begin{bmatrix} 0.125 \\ 0.156 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0.094 \end{bmatrix}, \quad \sigma = I + AI \dots + A^5 I = \begin{bmatrix} 3.938 \\ 3.531 \\ 2.5 \\ 1.5 \\ 1.5 \\ 1 \\ 3.656 \end{bmatrix}$$

(b) $\sigma = I + AI + \dots A^5 I$

Fig. 3. An example with propagation probability $p = 0.5$. Fig. 3(a) shows the adjacent matrix where $A_{ij} = 0.5$ means there is a directed edge from i to j. Fig. 3(b) shows how to calculate σ.

6 Algorithm for The Tree Network

In this section, we consider MIR problem on a special network such as a tree. The reason is as follows: (1) Calculating the activation probability of a node is very easy in a tree network; (2) We also show that, in a tree, the MIR problem can be solved by using an efficient dynamic programming method. And we believe that finding blockers in a tree can improve the understanding of information propagation process.

6.1 Calculating Activation Probability

In this part, we will show how to compute the activation probability for a node v by the rumor seed set S when the input graph is a tree. Notice that we have given the activation probability calculation formula for a node on the general graph in the previous section, i.e., equation (1). We modify the formula (1) as follow.

Definition 4. Given a directed tree $T = (V, E, p)$, a rumor seed set $S \subseteq V$ and the IC model $\mathcal{M}$, for any a node $v \in T$, let $Pr_{\mathcal{M}}(v, S)$ be the activation probability that the node v is activated by the seed set S. Then we have

$$Pr_{\mathcal{M}}(v,S) = \begin{cases} 1, & \text{if } v \in S \\ 0, & \text{if any } s \in S \text{ cannot reach } v \\ 1 - \prod_{s \in S} (1 - \prod_{(y,z) \in path(s,v)} p_{yz}), & \text{otherwise.} \end{cases} \tag{5}$$

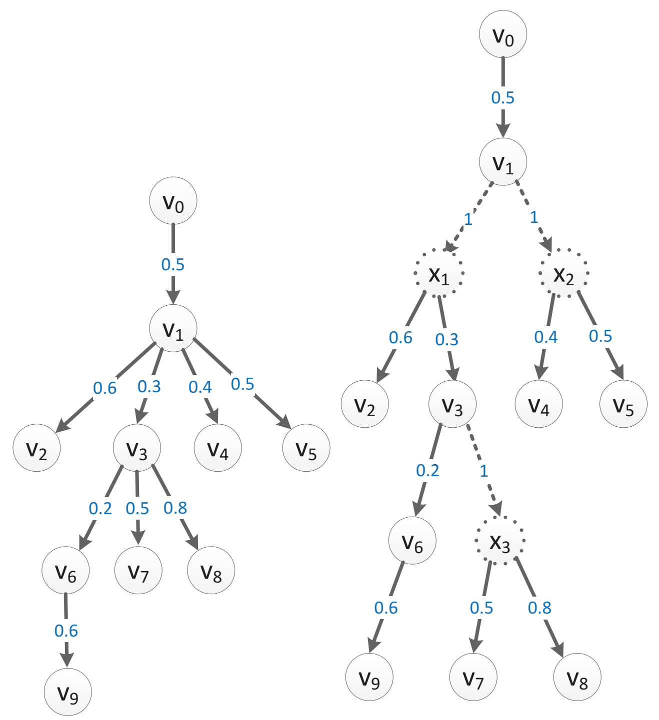
Where $path(s, v)$ denotes the path from the node s to the node v.

A node v on the tree is activated depending on whether there is a directed path $path(s, v)$ from the node s in the seed set to the node v. The term $\prod_{s \in S} (1 - \prod_{(y,z) \in path(s,v)} p_{yz})$ indicates the probability that all nodes in the seed set cannot activate the node v. Therefore the probability that the node v can be activated is equal to 1 minus this term.

6.2 Dynamic Programming Algorithm

In this subsection, we explore the MIR problem on a directed tree $T = (V, E, p)$ when the rumor seed set S is given. The key issue is the following: How to pick blocker nodes on the subtrees that the root nodes are the seed nodes. In order to address this problem, we propose a dynamic programming algorithm including two steps: (1) Convert a general tree into a binary tree; (2) Implement dynamic programming.

Convert a general tree into a binary tree. Let T_v be a subtree. And the root of T_v is the node v and v has θ children. The optimal way of specifying at most k' ($k' \leq k$) blockers on this subtree is to allocate k' blockers on this subtree to minimize the total activation probability. However, computing the all possible allocations is expensive when the $\theta \gg 2$. To solve this problem, we convert original three T_v rooted at v into a binary tree BT_v rooted at v. In addition, we will show this transformation is beneficial to formulate the dynamic programming.



(a) The general tree T_{v_0} (b) The binary tree BT_{v_0}

Fig. 4. The transformation from a general tree to a binary tree. Fig. 4(a) shows the general tree T_{v_0} and Fig. 4(b) shows the corresponding binary tree BT_{v_0} after transformation.

In Fig. 4, we give an example to show the transformation process. Fig. 4(a) shows a general tree T_{v_0} rooted at v_0 and Fig. 4(b) shows the corresponding binary tree BT_{v_0} rooted at v_0 after transformation where the dashed arrows and the dashed nodes mean the newly added directed edges and nodes. We make a simple transformation as follow:

- The root node v_0 of the general tree T_{v_0} is still the root node v_0 on the binary tree BT_{v_0};
- Suppose that v is any node of the general tree T_{v_0} with θ children y_1, y_2,..., y_θ. We replace the node v

with a binary tree with a height of no more than $\log_2\theta$ and the leaf nodes of this binary tree are y_1, y_2,..., y_θ. Perform the above replacement process on the general tree for the nodes whose out-degree is greater than 2.

In Fig. 4(a), the root node v_0 of the general tree T_{v_0} is still the root node on the binary tree BT_{v_0} in Fig. 4(b). We observe that the node v_1 has four children v_2, v_3, v_4 and v_5.(e.g., $\theta = 4$ is greater than 2). Therefore, we replace the node v_1 with a binary tree with a height of no more than $\log_2 4 = 2$ and the leaf nodes of this binary tree are v_2, v_3, v_4 and v_5[3]. In this case, we add two dashed nodes x_1 and x_2. And we also add two dashed directed edges (v_1, x_1) and (v_1, x_2). We let the propagation probability from the node v_1 to x_1 be 1. In addition, the solid directed edges should be retained, e.g., the directed edge (v_1, x_1) is replaced by the directed edge (x_1, v_2). A similar operation can be applied to the other edges such as (v_1, v_3), (v_1, v_4) and (v_1, v_5). Furthermore, we do a similar transformation to the node v_3 since it has three children. Finally, the binary tree is shown in Fig. 4(b).

Lemma 1. Given a general tree $T = (V, E, p)$, for any node $v \in V$ with θ children y_1, y_2,..., y_θ, then the height of the binary tree BT_v replacing the node v does not exceed $\log_2\theta$.

Proof 2. We assume that v is the 0-th layer of the binary tree BT_v, then the 1-th layer is at most two leaf nodes, the 2-th layer is at most four leaf nodes, and so on. Therefore, the $\log_2\theta$ layer has θ leaf nodes.

Lemma 2. Let $|V|$ be the number of nodes on the general tree $T = (V, E, p)$ and $|V'|$ be the number of all nodes on corresponding binary tree $BT = (V', E', p')$, respectively. Then we have $|V'| \le 3|V|$.

Proof 3. For ease of exposition, we divide the nodes in the general tree $T = (V, E, p)$ into two categories: (1) The set D, that is, the nodes whose the number of children is less or equal to 2; (2) The set $V - D$, that is, the nodes whose the number of children is greater than 2. For the first category, we let $|D|$ denote the size of set D. And we don't need a replacement operation, so the number of such nodes is the same. For the second category, we let $|V - D|$ denote the size of set $V - D$. However, we need replacement operations. Based on the above analysis, we have

$$\begin{aligned}|V'| &= |D| + (2^0 + ... + 2^{log_2\theta_1}) + (2^0 + ... + 2^{log_2\theta_2} + ...\\&\quad + (2^0 + ... + 2^{log_2\theta_{|V-D|}}))\\&\le |D| + \underbrace{2\theta_1 + 2\theta_2 + ... + 2\theta_{|V-D|}}_{|V-D|\ terms}\\&\le |V| + 2(\theta_1 + \theta_2 + ... + \theta_{|V-D|})\\&\le |V| + 2|V| = 3|V|.\end{aligned}$$

We assume that the nodes in the $V - D$ have θ_1, θ_2,..., $\theta_{|V-D|}$ children, respectively.

3. It is worth mentioning that the result (binary tree) of this transformation may not be unique, but we will show that all binary trees have the same optimal solution for the MIR problem.

Implement dynamic programming. Recall that we have a blocker budget k. The essence of our dynamic programming algorithm is to study how to select the blockers on the binary tree to minimize the objective function (total activation probability). Let $l(v)$ or $r(v)$ be the left or right child of v where $v \in BT$. Let $OPT(v, \mathcal{B}, k)$ the probability in the subtree rooted at node v with k blockers and $\mathcal{B}$ keeping the blockers in the current solution. The optimal way of selecting k blocker from this subtree must belong to one of the following two patterns: (1) Selecting the root node of this subtree and $k - 1$ nodes in the children nodes (including root node); (2) Selecting the k nodes in the child nodes (not including root node). Based on the above analysis, we propose the following dynamic programming for a node $v \in BT$ when the rumor seed set is S.

$$\begin{aligned}OPT(v, \mathcal{B}, k) = \min\{&\min_{k'=0}^{k}\{OPT(l(v), \mathcal{B}, k')\\&+OPT(r(v), \mathcal{B}, k - k') + Pr(v, S, \mathcal{B}) \cdot I_v^1\},\\&Pr(v_3, S, \mathcal{B} \cup \{v\}) \cdot I_v^2 + \min_{k'=0}^{k-1}\{OPT(l(v), \mathcal{B} \cup \{v\}, k')\\&+OPT(r(v), \mathcal{B} \cup \{v\}, k - k' - 1)\}\}.\end{aligned} \tag{6}$$

In equation (6), the term $\min_{k'=0}^{k}\{\cdot\}$ denotes that we do not select the root node but select k blockers from the child nodes of the root node. And the term $Pr(v, S, \mathcal{B})$ denotes the probability that the node v is activated by the seed set S. We define the indicator variable I_v^1 as

$$I_v^1 = \begin{cases} 1, & \text{if } v \notin S \text{ and } v \text{ is not a dashed node.}\\ 0, & \text{if } v \in S \text{ or } v \text{ is a dashed node.}\end{cases}$$

The term $\min_{k'=0}^{k-1}\{\cdot\}$ denotes that we select the root node and select $k - 1$ blockers from the child nodes. In order to ensure that the newly added dashed nodes cannot be selected, we define the indicator variable I_v^2 as

$$I_v^2 = \begin{cases} 0, & \text{if } v \notin S \text{ and } v \text{ is not a dashed node.}\\ \infty, & \text{if } v \in S \text{ or } v \text{ is a dashed node.}\end{cases}$$

From the above dynamic programming, we have the following two corollary.

Corollary 1. Dynamic programming can find the optimal solution of the MIR problem and the optimal solution on the binary tree is equivalent to the optimal solution on the general tree.

6.3 Case Study

In this subsection, we provide a case study when the budget $k = 2$. In Fig.4, without loss of generality, we let the node v_3 be the rumor seed node, i.e., $S = \{v_3\}$. Let us first analyze the optimal solution on the original thee, and then use the dynamic programming method to analyze the optimal solution on the binary tree.

On the original tree, the seed node v_3 is able to activate n-odes v_6, v_7, v_8 and v_9 with probabilities 0.2, 0.5, 0.8 and 0.12. Therefore we should select the node v_7 and node v_8 such that total probability $\sum_{v \in V \setminus \{S \cup \mathcal{B}\}} Pr_{\mathcal{C}}(v, S) = 0.2 + 0.12 = 0.32$ is minimized. In other words, the optimal solution of MIR problem is $\mathcal{B} = \{v_7, v_8\}$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2019.2903272, IEEE Transactions on Network Science and Engineering

According to dynamic programming, our goal is

$$\begin{aligned} OPT(v_3, \mathcal{B}, 2) = \min\{ \\ \min_{k'=0}^{2}\{OPT(l(v_3), \mathcal{B}, k') + OPT(r(v_3), \mathcal{B}, 2-k') \\ +Pr(v_3, S, \mathcal{B}) \cdot I^1_{v_3}\}, \min_{k'=0}^{1}\{OPT(l(v_3), \mathcal{B} \cup \{v_3\}, k') \\ +OPT(r(v_3), \mathcal{B} \cup \{v_3\}, 2-k'-1) \\ +Pr(v_3, S, \mathcal{B} \cup \{v_3\}) \cdot I^2_{v_3}\}\}. \end{aligned}$$

The node v_3 cannot be selected into the blocker set $\mathcal{B}$ because v_3 is the rumor seed node. Thus we have

$$\begin{aligned} OPT(v_3, \mathcal{B}, 2) = \min_{k'=0}^{2}\{OPT(l(v_3), \mathcal{B}, k') \\ + OPT(r(v_3), \mathcal{B}, 2-k') + Pr(v_3, S, \mathcal{B}) \cdot I^1_{v_3}\}, \end{aligned}$$

where the $Pr(v_3, S, \mathcal{B}) \cdot I^1_{v_3}$ is 0 since $I^1_{v_3} = 0$ when v_3 is the rumor seed node. Then we have

$$\begin{aligned} OPT(v_3, \mathcal{B}, 2) = \\ \min_{k'=0}^{2}\{OPT(l(v_3), \mathcal{B}, k') + OPT(r(v_3), \mathcal{B}, 2-k')\}. \end{aligned} \tag{7}$$

Based on equation (7), we have the following three cases:

- **Case 1:**$OPT(l(v_3), \mathcal{B}, 0) + OPT(r(v_3), \mathcal{B}, 2)$;
- **Case 2:**$OPT(l(v_3), \mathcal{B}, 1) + OPT(r(v_3), \mathcal{B}, 1)$;
- **Case 3:**$OPT(l(v_3), \mathcal{B}, 2) + OPT(r(v_3), \mathcal{B}, 0)$;

In **Case 1**, on one hand, $OPT(l(v_3), \mathcal{B}, 0) = 0.2 + 0.12 = 0.32$ since we do not select any node as a blocker on the left subtree. On the other hand,

$$\begin{aligned} OPT(r(v_3), \mathcal{B}, 2) = OPT(x_3, \mathcal{B}, 2) = \min\{ \\ \min_{k''=0}^{2}\{OPT(l(x_3), \mathcal{B}, k'') + OPT(r(x_3, \mathcal{B}, 2-k'')) \\ +Pr(x_3, S, \mathcal{B}) \cdot I^1_{x_3}\}, \min_{k''=0}^{1}\{OPT(l(x_3), \mathcal{B} \cup \{x_3\}, k'') \\ +OPT(r(x_3), \mathcal{B} \cup \{x_3\}, 1-k'') + Pr(x_3, S, \mathcal{B}) \cdot I^2_{x_3}\}\}. \end{aligned} \tag{8}$$

In equation (8), $Pr(x_3, S, \mathcal{B}) \cdot I^1_{x_3} = 0$ because the node x_3 is dashed node and we do not select this node in our optimal solution. Therefore we have

$$\begin{aligned} OPT(x_3, \mathcal{B}, 2) = \\ \min_{k''=0}^{2}\{OPT(l(x_3), \mathcal{B}, k'') + OPT(r(x_3, \mathcal{B}, 2-k'')). \end{aligned} \tag{9}$$

Similarly, we recursively seek the optimal solution of equation (9) and consider following three subcases:

i. $OPT(l(x_3), \mathcal{B}, 0) + OPT(r(x_3, \mathcal{B}, 2)$;
ii. $OPT(l(x_3), \mathcal{B}, 1) + OPT(r(x_3, \mathcal{B}, 1)$;
iii. $OPT(l(x_3), \mathcal{B}, 2) + OPT(r(x_3, \mathcal{B}, 0)$;

However, the subcases (i) and (iii) do not exist because the number of nodes available for selecting on right or left subtree of x_3 is less than k. Thus the optimal solution of (9) is $OPT(l(x_3), \mathcal{B}, 1) + OPT(r(x_3, \mathcal{B}, 1) = 0 + 0 = 0$. And it indicates that the blocker set $\mathcal{B} = \{v_7, v_8\}$. In summary, in **Case 1**, $\{OPT(l(v_3), \mathcal{B}, 0) + OPT(r(v_3), \mathcal{B}, 2) = 0.32 + 0 = 0.32$ and the blocker set $\mathcal{B} = \{v_7, v_8\}$.

Analogously, we analyze the **Case 2** and the **Case 3**, respectively. Here, we omit the detailed calculation process and give the results directly. In **Case 2**, $OPT(l(v_3), \mathcal{B}, 1) + OPT(r(v_3), \mathcal{B}, 1) = 0 + 0.5 = 0.5$ and the blocker set $\mathcal{B} = \{v_6, v_8\}$. In **Case 3**, $OPT(l(v_3), \mathcal{B}, 2) + OPT(r(v_3), \mathcal{B}, 0) = 0 + 0.5 + 0.8 = 1.3$ and the blocker set $\mathcal{B} = \{v_6, v_9\}$.

Finally, comparing **Case 1**, **Case 2** and **Case 3**, we observe that the optimal solution of MIR problem is $\mathcal{B} = \{v_7, v_8\}$ since the total activation probability is minimized. In addition, we also find that the result of the dynamic programming algorithm in the binary tree is equal to the optimal solution in general tree.

7 EXPERIMENT

In this section, we evaluate proposed algorithm on synthetic and real-life networks. First, we describe the data sets and experiment setup. Second, we analyze and discuss experimental results from different perspectives. Finally, we compare with other heuristic approaches.

7.1 Data sets

The experimental datasets are divided into two categories: the synthetic networks and the real-life networks. More specifically, for the random networks, we generate a general network SYN and a tree network SYN-T. For the real-life networks, we collect three datasets with various scale from *Stanford Large Network Dataset Collection* (SNAP)[4] and the *Koblenz Network Collection* (KONECT)[5], respectively. Table 1 provides the details of these data sets. In table, "CC" represents clustering coefficient and "MD" represents the maximum degree.

- **Synthetic** (SYN). We randomly generate a graph using Erdos-Renyi model [24] which assigns equal probability η to all nodes. The higher assigned probability is, the more dense the graph is. In experiments, we let $\eta = 0.5$.
- **Synthetic Tree** (SYN-T). We randomly generate a tree network containing 1000 nodes and 999 directed edges. Note that each directed edge is from the parent node to the child node.
- **Wiki Vote** (WV). This network contains all the Wikipedia voting data from the inception of Wikipedia till January 2008. Nodes in the network represent wikipedia users and a directed edge from node u to node v represents that user u voted on user v.
- **Twitter Lists** (TL). This directed network contains Twitter user-user following information. A node represents a user. An edge (u, v) indicates that the user u follows the user v.
- **Google+** (G+). This directed network contains Google+ user-user links. A node represents a user, and a directed edge denotes that one user has the other user in his circles.

7.2 Experiment Setup

We make the following setup for rumor spreading process: Given a directed social network $G = (V, E, p)$, 1% of nodes

4. http://snap.stanford.edu/data
5. http://konect.uni-koblenz.de

TABLE 1
The details of synthetic and real-life social data sets.

Data Sets	Relationship	#Node	#Edge	CC	MD	Diameter
Synthetic	Synthetic	2000	10000	-	62	6
Synthetic tree	Synthetic	1000	999	-	38	10
Wiki Vote	Voting	7,115	103,663	0.14	875	7
Twitter Lists	Following	23,370	33,101	0.02	239	15
Google+	Friend sharing	23,628	39,242	0.03	2,771	8

are selected randomly and uniformly from V as rumor source set S. In our all experiments, we adopt *Independent Cascade* (IC) model as information diffusion model. In particular, we assign p in two ways since the data sets lack propagation probability p. One assigns a uniform probability $p = 0.5$ for each edge on the networks. Another assigns a trivalency model $p = TRI$ for each edge, i.e., we uniformly select a value from $\{0.1,0.01,0.001\}$ at random that corresponds to high, medium and low propagation probabilities. Notice that all networks are simple networks[6].

7.2.1 Comparison Methods:

To compare with existing methods, other heuristic methods such as **Out-Degree**, **Betweenness Centrality** and **PageRank** are selected as comparison methods. Our two-stage approach is abbreviated as GCSSB.

- **Out-Degree** (OD) [18]. The out-degree of a node v is the number of outgoing edges from the node v. Kempe et al. show high degree nodes may outperform other centrality-based heuristics in terms of influential identification.
- **Betweenness Centrality** (BC) [25]. A node's betweenness is equal to the number of shortest paths from all nodes to all others that pass through that node. Recently, betweenness centrality has become an important centrality measure in social networks.
- **PageRank** (PR) [26]. This is widely known Google Page-Rank measure. The pagerank score indicates the importance of a node. There is a damping factor parameter and we set it to 0.9 in all experiments.

7.2.2 Evaluation Criteria:

The experimental evaluation is carried out from the following aspects: (1) parameter α study. In our GCSSB method, we need to generate candidate set $\mathcal{C}$ with $\alpha * k$ nodes. (2) parameter k study. We study the relationship between the size of the blocker set and the objective function value. (3) Compare with other methods. We compare GCSSB with other heuristic methods such as Out-Degree, Betweenness Centrality and PageRank. Our evaluation criteria is objective function value (total activation probability). A smaller function value indicates that the algorithm is better. (4) Finally, we evaluate the proposed dynamic programming algorithm on the synthesized tree dataset.

7.3 Results

Parameter α study: We study the effect of candidate set size (parameter α) on the objective function value (total activation probability). The experimental results are shown in Fig. 5. Fig. 5(a) and Fig. 5(b) show the propagation probability $p = 0.5$ and $p = TRI$, respectively. The results on each network show the same trends in both subgraphs. Taking Fig. 5(a) as an example, the horizontal axis and the vertical axis represent the parameter α and the total activation probability, respectively. The total activation probability decreases as the parameter α increases. In particular, the total acceptance probability remains essentially the same when $\alpha \geq 6$. Therefore we let $\alpha = 6$ in latter experiments.

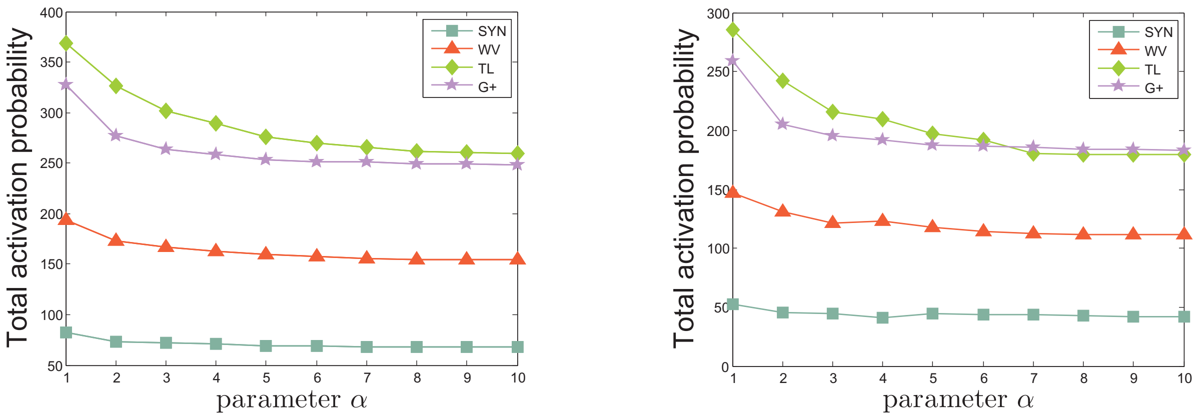
Parameter k study: We study the relationship between the size of the blocker set and the total activation probability. The experimental results are shown in Fig. 6. In the subfigures, the horizontal and vertical axes represent the parameter k and the total activation probability, respectively. Through experiments, we observe that the total activation probability decreases as k increases. In particular, it is drastically reduced when $k > 300$ on each network.

Comparing with other methods: We compare our GCSS-B with other methods (OD, BC and PR). The experimental results are shown in Fig. 7. The horizontal and vertical axes represent the parameter k and the total activation probability, respectively. In both subfigures, The total activation probability decreases as k increases. We observe that the proposed method is the best since the total activation probability is the smallest. Moreover, in comparison methods, the PR's performance is the best but the OD is the worst.

Evaluating dynamic programming algorithms in tree. We evaluate the proposed dynamic programming algorithm in the tree network SYN-T. We first randomly and uniformly select 40 nodes as the rumor seed nodes on the tree network SYN-T. Then we employ the independent cascade model as information propagation model and select the blocker sets by the dynamic programming in two propagation probability ways. The results are shown in Fig. 8. In the figure, the horizontal axis indicates the parameter k varying from 0 to 100 and the vertical axis indicates the total activation probability. In the both subfigures, we compare the activation probability returned by dynamic programming (DP) with the one without any blockers (Baseline). We have the following observations: (1) The total activation probability decreases as the parameter k increases. This phenomenon once again proves that the goal is monotonously decreasing. In other words, the more blockers we select, the smaller the total activation probability should be; (2) When $k < 40$, the total activation probability is reduced sharply. However, when $k > 40$, the total activation probability is reduced steadily. Our dynamic programming automatically selects the optimal nodes as the blockers.

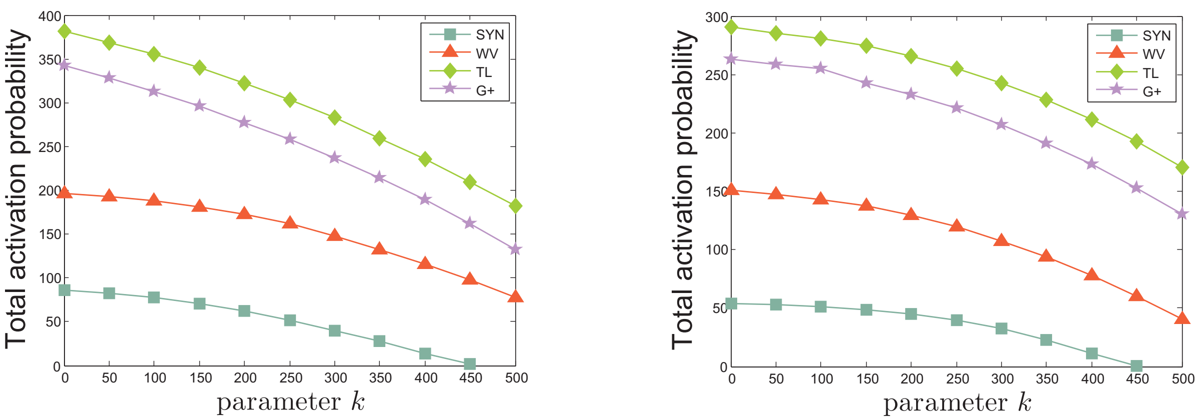
6. Self-loops and multiple edges are not allowed.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2019.2903272, IEEE Transactions on Network Science and Engineering



(a) $p = 0.5$ (b) $p = TRI$

Fig. 5. Total activation probability Vs. Parameter α under IC model: rumor source $|S| = 1\%|V|$ on each network, $p = 0.5$ or $p = TRI$, and $k = 50$.



(a) $p = 0.5$ (b) $p = TRI$

Fig. 6. Total activation probability Vs. Parameter k under IC model: rumor source $|S| = 1\%|V|$ on each network, $p = 0.5$ or $p = TRI$, and $\alpha = 6$.

8 Conclusions

In this paper, we study a novel problem called *Minimizing Influence of Rumor* (MIR) problem that finds a small size blocker set such that the activation probability of users on network is minimized. Based on IC model, we prove objective function satisfies non-submodularity. We develop a two-stages method GCSSB to quickly identify blocker set in the general networks. Furthermore, we propose a dynamic programming algorithm in the tree networks and find it can provide the optimal solution. Finally, in order to evaluate our proposed methods, extensive experiments have been conducted. The experiment results show that our method outperforms comparison approaches.

Acknowledgments

This work is partly supported by National Natural Science Foundation of China under grant 11671400, 61672524 and National Science Foundation under grant 1747818.

References

[1] B. Doerr, M. Fouz, and T. Friedrich, "Why rumors spread so quickly in social networks," *Communications of the ACM*, vol. 55, no. 6, pp. 70–75, 2012.

[2] S. Wen, J. Jiang, Y. Xiang, S. Yu, W. Zhou, and W. Jia, "To shut them up or to clarify: Restraining the spread of rumors in online social networks," *IEEE Transactions on Parallel & Distributed Systems*, vol. 25, no. 12, pp. 3306–3316, 2014.

[3] L. Fan, Z. Lu, W. Wu, B. Thuraisingham, H. Ma, and Y. Bi, "Least cost rumor blocking in social networks," in *Distributed Computing Systems (ICDCS), 2013 IEEE 33rd International Conference on*. IEEE, 2013, pp. 540–549.

[4] S. Wang, X. Zhao, Y. Chen, Z. Li, K. Zhang, and J. Xia, "Negative influence minimizing by blocking nodes in social networks." in *AAAI (Late-Breaking Developments)*, 2013, pp. 134–136.

[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.

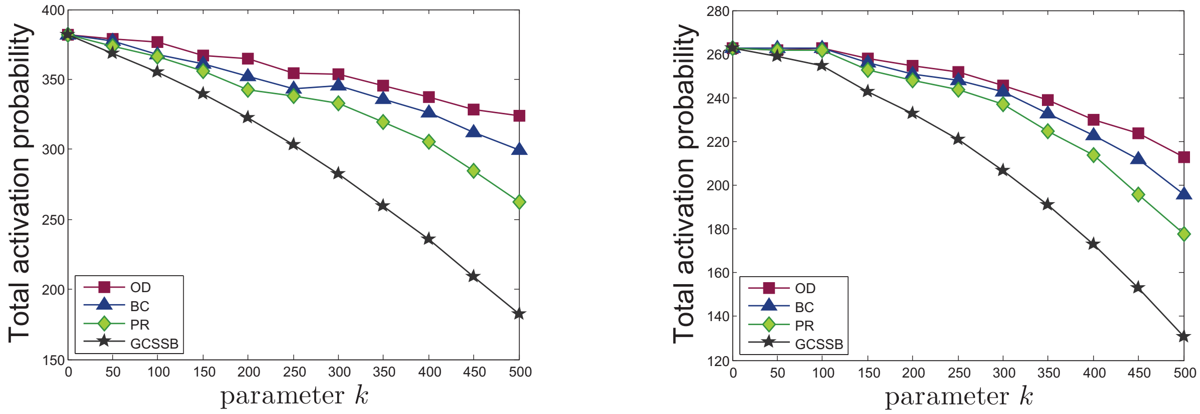
[6] C. Gao, J. Liu, and N. Zhong, "Network immunization and virus propagation in email networks: experimental evaluation and analysis," *Knowledge & Information Systems*, vol. 27, no. 2, pp. 253–279, 2011.

[7] C. Reuven, H. Shlomo, and B. A. Daniel, "Efficient immunization strategies for computer networks and populations," *Physical Review Letters*, vol. 91, no. 24, p. 247901, 2003.

[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.

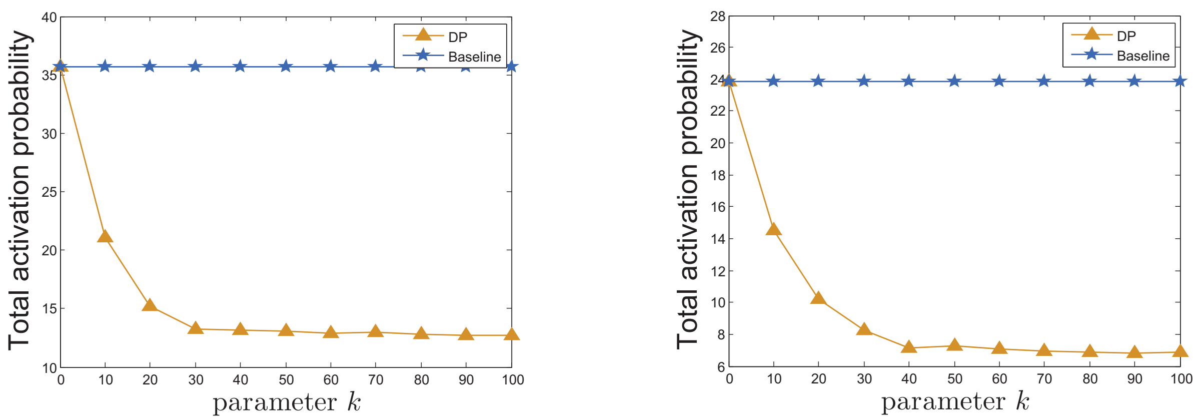
[9] E. B. Khalil, B. Dilkina, and L. Song, "Scalable diffusion-aware optimization of network topology," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014, pp. 1226–1235.

[10] 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*



(a) $p = 0.5$, TL network (b) $p = TRI$, G+ network

Fig. 7. Compare with other methods: rumor source $|S| = 1\%|V|$, $p = 0.5$ on TL network or $p = TRI$ on G+ network, and $\alpha = 6$.



(a) $p = 0.5$, SYN-T network (b) $p = TRI$, SYN-T network

Fig. 8. Evaluating dynamic programming algorithm: rumor source $|S| = 40$, $p = 0.5$ or $p = TRI$ on SYN-T tree network.

and knowledge management. ACM, 2012, pp. 245–254.

[11] T. Nepusz and T. Vicsek, "Controlling edge dynamics in complex networks," *Nature Physics*, vol. 8, no. 8, pp. 568–573, 2011.

[12] N. P. Nguyen, X. Ying, and M. T. Thai, "A novel method for worm containment on dynamic social networks," in *Military Communications Conference, 2010 - Milcom*, 2010, pp. 2180–2185.

[13] 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.

[14] 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.

[15] C. Gao and J. Liu, "Modeling and restraining mobile virus propagation," *IEEE Transactions on Mobile Computing*, vol. 12, no. 3, pp. 529–541, 2013.

[16] S. Shirazipourazad, B. Bogard, H. Vachhani, A. Sen, and P. Horn, "Influence propagation in adversarial setting: how to defeat competition with least amount of investment," in *Acm International Conference on Information & Knowledge Management*, 2012, pp. 585–594.

[17] P. Domingos and M. Richardson, "Mining the network value of customers," in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2001, pp. 57–66.

[18] 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.

[19] Y. Zhu, D. Li, and Z. Zhang, "Minimum cost seed set for competitive social influence," in *INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications, IEEE*. IEEE, 2016, pp. 1–9.

[20] R. Yan, Y. Zhu, D. Li, and Z. Ye, "Minimum cost seed set for threshold influence problem under competitive models," *World Wide Web*, pp. 1–20, 2018.

[21] S. Wen, M. S. Haghighi, C. Chen, Y. Xiang, W. Zhou, and W. Jia, "A sword with two edges: Propagation studies on both positive and negative information in online social networks," *IEEE Transactions on Computers*, vol. 64, no. 3, pp. 640–653, 2015.

[22] T. Lappas, E. Terzi, D. Gunopulos, and H. Mannila, "Finding effectors in social networks," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2010, pp. 1059–1068.

[23] S. Bharathi, D. Kempe, and M. Salek, *Competitive Influence Maximization in Social Networks*. Springer Berlin Heidelberg, 2007.

[24] P. ERDdS and A. R&WI, "On random graphs i," *Publ. Math. Debrecen*, vol. 6, pp. 290–297, 1959.

[25] U. Brandes, "On variants of shortest-path betweenness centrality and their generic computation," *Social Networks*, vol. 30, no. 2, pp. 136–145, 2008.

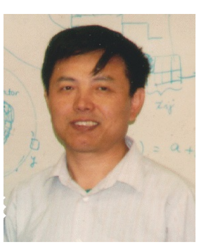
[26] L. Page, S. Brin, R. Motwani, T. Winograd *et al.*, "The pagerank citation ranking: Bringing order to the web," 1998.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2019.2903272, IEEE Transactions on Network Science and Engineering

Ruidong Yan received the BS degree in Information and Computing Sciences from Inner Mongolia University, Hohhot, China(2014), He is a PhD candidate in the Department of Computer Science at Renmin University of China. His research interests include social networks, algorithm design and analysis.

Deying Li is a professor of Renmin University of China. She received the B.S. degree and M.S. degree in Mathematics from Huazhong Normal University, China, in 1985 and 1988 respectively. She obtained the PhD degree in Computer Science from City University of Hong Kong in 2004. Her research interests include wireless networks, ad hoc & sensor networks mobile computing, distributed network system, social networks, and algorithm design etc.

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 current research interests include data communication, data management, the design and analysis of algorithms for optimization problems that occur in wireless networking environments, and various database systems.



Ding-Zhu Du received the M.S. degree from the Chinese Academy of Sciences in 1982 and the Ph.D. degree from the University of California at Santa Barbara in 1985, under the supervision of Professor Ronald V. Book. Before settling at the University of Texas at Dallas, he worked as a professor in the Department of Computer Science and Engineering, University of Minnesota. He also worked at the Mathematical Sciences Research Institute, Berkeley, for one year, in the Department of Mathematics, Massachusetts Institute of Technology, for one year, and in the Department of Computer Science, Princeton University, for one and a half years. He is the editor-in-chief of the Journal of Combinatorial Optimization and is also on the editorial boards for several other journals. Forty Ph.D. students have graduated under his supervision. He is a member of the IEEE.

Yongcai Wang received his BS and PhD degrees from department of automation sciences and engineering, Tsinghua University in 2001 and 2006 respectively. He worked as associated researcher at NEC Labs. China from 2007-2009. He was an research scientist in Institute for Interdisciplinary Information Sciences, Tsinghua University from 2009-2015. He was a visting scholar at Cornell University in 2015. He is currently associate professor at Department of Computer Sciences, Renmin University of China. His current research interests include network localization algorithms, internet of things, combinatorial optimization and applications.