Computer Communications 182 (2022) 41–51

Contents lists available at ScienceDirect

Computer Communications

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

Fast controlling of rumors with limited cost in social networks

Xiaopeng Yao, Yue Gu, Chonglin Gu, Hejiao Huang *

Harbin Institute of Technology, Shenzhen, China

ARTICLE INFO

Keywords:
Rumor
Social network
Limited cost
Contact coefficient

ABSTRACT

Innumerable rumors spread quickly through social networks and how to fast control the spread of rumors is crucial. An efficient way is to take different measures for users with different influence extent by rumors, but the costs of these measures vary. In this paper, we try to minimize the spread (termination) time of rumors considering the controlling cost. To solve this problem, we creatively design five different measures to control rumors. First, we propose a contact coefficient to quantify the influence weight for each user. Second, we classify the users to different groups based on their influence weights so that the rumors can be controlled by the measures accordingly. Thus, the controlling of rumors can be formulated into an optimization problem, in which decision variables based on contact coefficients are used to classify users. Then an approximation algorithm named WB-GA is designed to classify different users, ensuring that rumors can be controlled as fast as possible within given costs. The experimental results on real online networks show that our algorithm is highly efficient and effective.

1. Introduction

With the growing popularity of online social networks, information can spread faster and more widely than ever before. Meanwhile, the establishment of such an environment facilitates the spreading of rumors. Rumors are often defined as unverified statements [1,2] that might be eventually found as true or false. For example, a disaster swept the whole world and there came a lot of rumors, which can spread much faster than the disaster itself, causing more damage. It is conceivable that there will be huge economic losses and even society instability if not stopping the spreading of rumors in time. Therefore, fast controlling of rumors is urgent and necessary in social networks.

In literature, the commonly used methods to control the rumors can be divided into three categories: (1) Removing associations between users to block rumors [3–5]; (2) Blocking influential users [6–8]; (3) Spreading truth to clarify rumors [9–11]. In addition, B. Wang et al. [12] in a recent work introduce experiences into rumor controlling, and G. Tong et al. [13] propose a fast randomized approximation to control rumors. However, all of these studies just take one measure in rumor controlling.

In reality, the extent that users are influenced by rumors can be quite different, and the best way to control the rumors is to take different measures for the users. For example, for those users who are influenced by rumors or even want to continue to spread rumors, we should stop this process by deleting their accounts. For those users who have a high probability of being influenced by rumors, we should spread truth to them or block them from accessing information. For those users who have a low probability of being influenced by rumors, we just need to tag and track them. In view of this, we propose to classify the users into different groups with different controlling measures. To be practical, we rank users from lowest to highest probability of being influenced by rumors and we consider classifying users into 5 groups (H_1, H_2, H_3, H_4, H_5) with measures of *taking no action*, *tagging the user*, *blocking access to information*, *spreading the truth*, *deleting the user's account*, respectively. Note that, the cost of these measures are different. We aim to use different control measures with limited cost to control rumors as fast as possible. In order to quantify the influence weight for each user, we use a contact coefficient to classify the users.

In this paper, we study the problem of fast controlling of rumors (called *FCR* problem) in a social network with limited cost by using different controlling measures. Note that, the spread time of rumors is defined as the time when all rumors terminate in the spreading process. Given the total cost, our goal is to minimize the spread time of rumors under the constraint of cost. The decision variables are the classifying boundaries of each group based on the contact coefficient. The classifying boundary for each group refers to the maximum and minimum contact coefficients of users in each group.

The main contributions of this paper are presented as follows:

• We creatively propose to classify users to different groups with five measures for rumor controlling, which is of great significance in practice. The five different measures are *taking no action*, *tagging the user*, *blocking access to information*, *spreading the truth* and *deleting the user's account*.

* Corresponding author.
E-mail address: huanghejiao@hit.edu.cn (H. Huang).

https://doi.org/10.1016/j.comcom.2021.10.041
Received 13 January 2021; Received in revised form 20 September 2021; Accepted 29 October 2021
Available online 6 November 2021
0140-3664/© 2021 Elsevier B.V. All rights reserved.

• We propose a Multi-Probability Independent Cascade (MPIC) model to describe the process of spreading rumors. We also propose a well-designed contact coefficient and model the controlling of rumors into an optimization problem.

• We design a Withdraw Bedeckung-Greedy Algorithm named as WB-GA to solve this problem. After a series of theoretical analysis, we prove that the data-parameter-based approximation ratio of the WB-GA is more than $\log_{(1-p_{(u,v)}+\alpha_3)}(1-p_{(u,v)}+\alpha_2)$, where $p_{(u,v)}$ is the influence probability for each edge without taking any measure, α_2 is the blocking influence rate of H_2 and α_3 is the blocking influence rate of H_3.

• Numerical experiments are conducted to evaluate the performance of our algorithm. Under the datasets of real online networks including Lastfm-social and Deezer-social, the performances of WB-GA are superior to that of the comparison methods.

The rest of this paper is organized as follows. Section 2 reviews related work. In Section 3, we formulate the controlling of rumors into an optimization problem. Section 4 give the details of our algorithm WB-GA. In Section 5, we conduct theoretical analysis of the problem and the algorithm. In Section 6, we make evaluations through extensive experiments. Finally, we conclude the whole paper in Section 7.

2. Related work

So far, there have been much work on rumor detection [14–17]. S. Kwon et al. [18] and J. Ma et al. [19] propose handcrafted-features based algorithms to detect rumors. F. Yu et al. [20] and J. Ma et al. [21] propose deep-learning based methods using Gated Recurrent Unit (GRU) and Convolutional Neural Network (CNN).

There are also a lot of scholars [22–24] studying the controlling of rumors. H. Jing et al. [25] and X. Liu et al. [26] try to identify a set of target nodes which will spread anti-rumor to counter the rumor. L. Yang et al. [9] propose a competitive diffusion model, namely Linear Threshold model with One Direction state Transition (LT1DT), and a novel heuristic based on diffusion dynamics is proposed to solve the problem of controlling rumors under the model. Z. Tan et al. [10] propose Activation Increment Minimization (AIM) strategy to select and block nodes for controlling rumors. A.I.E. Hosni et al. [11] propose a truth campaign strategy in minimizing the influence of rumors in multiplex online social networks from the perspective of network inference and by exploiting the survival theory. H. Zhang et al. [27] mainly focus on the limiting rumors with known sources case and propose an effective algorithm, exploiting the critical nodes.

Additionally, B. Wang et al. [12] try to block rumor based on user experience in real-world social networks. C.J. Kuhlman et al. [28] investigated contagion blocking in networked populations by identifying edges to remove from a network and P. Dey et al. [29] consider how to control rumor from content.

Moreover, many researchers propose a $(1-1/e)$-approximate algorithms to control rumors based on different cascade models. For example, C. Budak et al. [30] propose a competitive model, X. He et al. [31] propose the influence blocking maximization problem based on the competitive linear threshold model and L. Fan et al. [32] propose the opportunistic one-active-one model. However, these algorithms are too time consuming. Therefore, G. Tong et al. [24] propose a randomized approximation algorithm which is provably superior to the state-of-the art methods with respect to running time.

Different from the above work, other researchers try to dynamically control rumors. K. Scaman et al. [33] study and define the dynamic control of rumors as a dynamic resource allocation (DRA) problem and propose a novel Largest Reduction in Infectious Edges (LRIE) control strategy. A. Kalogeratos et al. [34] propose dynamic strategies for allocating resources on the nodes of an arbitrary network to control rumors.

Table 1
Notations.

Symbol	Definition
τ	The spread time of rumor
$p_{(u,v)}$	The influence probability from node u to node v for each edge $(u,v) \in E$ without taking any measure
$H_i (i \in \{1,2,3,4,5\})$	The set of nodes classified into group i
$H_{t,i} (i \in \{1,2,3,4,5\})$	The set of nodes in group i at time t
C_{total}	The total cost
v^{sta}	The state of node v
$C_i (i \in \{1,2,3,4,5\})$	The cost when taking measure i for a user
$\alpha_i (i \in \{1,2,3,4\})$	The blocking influence rate for each node in H_i

3. Problem model

In this section, we first introduce the preliminaries of this paper, and then give the problem model in detail. The notations and definitions to be used are listed in Table 1.

3.1. Preliminaries

Many scholars use the Independent Cascade (IC) model [35] to simulate the spread of rumors. In this model, given a social graph $G(V,E)$, where V denotes the set of nodes and E denotes the set of edges. Each node represents a user and each edge represents that there is a connection between the two users and it will influence its neighbor nodes with a certain probability. Each node can be influenced by rumors at most once [24,30–32] in the whole spreading process.

We use τ to denote the spread time of rumors and use t to denote the time. From a theoretical perspective, each user needs to be classified. Therefore, the purpose is to classify the group so that the probability of being influenced can be further calculated. Based on this classification, it is convenient to calculate the change of the probability of being influenced at the next moment. To be practical, we consider the following five controlling measures. Let $H_i (i \in \{1,2,3,4,5\})$ represent the set of users classified into group i that should take different measures. Here H_5 is *Deleting the user's account* set, representing that users' accounts in this group should be deleted. Similarly, H_4 is *Spreading the truth* set, H_3 is *Blocking access to information* set, H_2 is *Tagging the user* set, H_1 is *Taking no action* set. Moreover, let $H_{t,i}$ represent the set of users in H_i at time t. If a user is in H_5, he/she cannot be influenced by or spread rumor, while other users in H_1, H_2, H_3, H_4 can still be influenced by rumor but with different probabilities. In real life, the cost of these five measures can be different. We denote the cost of taking measure i for one user as C_i, and the given total cost as C_{total}.

The state of node v is defined as follows:

$$v^{sta} = \begin{cases} True, & v\ is\ influenced\ by\ rumors \\ False, & v\ is\ not\ influenced\ by\ rumors \end{cases} \tag{1}$$

3.2. Multi-probability independent cascade model

In this paper, we propose a MPIC model to describe the process of rumor spreading. Given a social network $G(V,E)$, we assume that the total number of users in this network is $|V| = n$. Let $p_{(u,v)}$ denote the influence probability from node u to node v for each edge $(u,v) \in E$ without taking any measure. We use R to denote the set of rumor seed nodes.

In this model, we assume that each rumor node can influence its neighbors once. We assume that there are four blocking influence rates at the same time when we taking five controlling measures. We use $\alpha_i (i \in \{1,2,3,4\})$ to represent the blocking influence rate for each node in $H_i (i \in \{1,2,3,4\})$. We define that $0 \le \alpha_1 < \alpha_2 < \alpha_3 < \alpha_4 < p_{(u,v)} \le 1$.

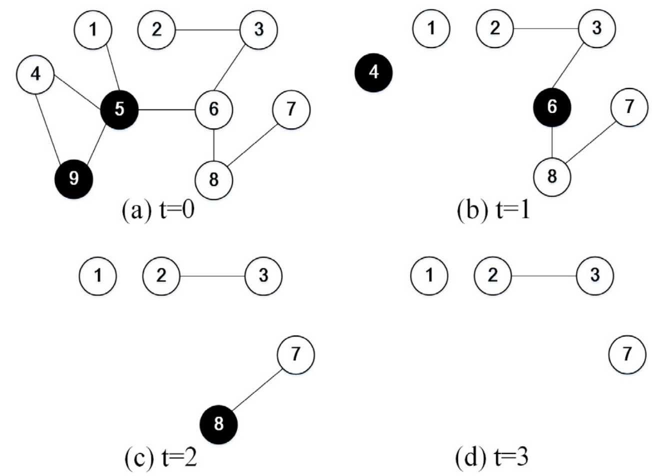

Fig. 1. An example of the spread of rumors under MPIC model.

When taking five controlling measures, the probability of each node v being influenced by its neighbors is as follows.

$$g_{(u,v)} = \begin{cases} p_{(u,v)} - \alpha_1, & v \text{ is in } H_1 \\ p_{(u,v)} - \alpha_2, & v \text{ is in } H_2 \\ p_{(u,v)} - \alpha_3, & v \text{ is in } H_3 \\ p_{(u,v)} - \alpha_4, & v \text{ is in } H_4 \end{cases} \tag{2}$$

The spread process of the MPIC model unfolds in discrete, as follows.

• Initially, at time $t = 0$, there are k rumor nodes (*True* nodes). Other $(n-k)$ nodes are *False* nodes and these nodes belongs to different $H_{0,i}(i = \{1,2,3,4\})$. Then the k *True* nodes will try to influence their neighbor nodes with different influence probabilities.

• At time $t \geq 1$. When a node u is influenced by rumors at time t, it can influence its neighbor node v with a influence probability $g_{(u,v)}$. If u succeeds, then v will become a *True* node at time $t+1$ and its state will never be changed again. Otherwise, node v can be influenced by other rumor nodes after time t.

• In the end, the spread process terminates when no node can be influenced by rumors.

Fig. 1 shows an example of the spread of rumors under MPIC model. In Fig. 1, there are 9 nodes and v_5, v_9 are *True* nodes at the initial time. Assume that the influence probability is 1 for all edges at the initial time and let $\alpha_1 = 0.1$, $\alpha_2 = 0.2$, $\alpha_3 = 0.3$, $\alpha_4 = 0.4$. We also assume that all nodes are already divided into different $H_{t,i}(i \in \{1,2,3,4,5\})$.

In Fig. 1(a), we assume that $H_{0,1} = \{v_2, v_7\}$, $H_{0,2} = \{v_3, v_8\}$, $H_{0,3} = \{v_1, v_6\}$, $H_{0,4} = \{v_4\}$ and $H_{0,5} = \{v_5, v_9\}$. Then, at time $t = 0$, v_5, v_9 will try to influence v_4 with the probability $(1 - \alpha_4 = 0.6)$ and v_5 will try to influence v_1, v_6 with the probability $(1 - \alpha_3 = 0.7)$. The other influence probabilities are $g_{(v_6,v_3)} = g_{(v_6,v_8)} = 1 - \alpha_2 = 0.8$, $g_{(v_8,v_7)} = g_{(v_3,v_2)} = 1 - \alpha_1 = 0.9$. In Fig. 1(b), at time $t = 1$, v_4, v_6 are successfully influenced by v_5. Then we delete the v_5, v_9 *(Deleting the user's account)* and let $H_{1,5} = \{v_4, v_6\}$. Let $H_{1,2} = \{v_2, v_7\}$, $H_{1,3} = \{v_3, v_8\}$ and $H_{1,1} = \{v_1\}$, $H_{1,4} = \emptyset$. Specially, since v_1 is an acnode, it would not be influenced by other nodes. Thus, v_6 will try to influence v_3, v_8 with the probability $(1 - \alpha_3 = 0.7)$. The other influence probabilities are $g_{(v_3,v_2)} = g_{(v_8,v_7)} = 1 - \alpha_2 = 0.8$. In Fig. 1(c), at time $t = 2$, v_8 is successfully influenced by v_6. Then we delete the v_4, v_6 and let $H_{2,5} = \{v_8\}$, $H_{2,3} = \{v_7\}$ and $H_{2,1} = \{v_1, v_2, v_3\}$, $H_{2,2} = H_{2,4} = \emptyset$. v_8 will try to influence v_7 with the probability $(1 - \alpha_3 = 0.7)$. v_1, v_2, v_3 would not be influenced by rumors. In Fig. 1(d), at time $t = 3$, v_7 is not influenced by v_8. We delete the v_8 and there is no node can be influenced by rumors. Therefore, the spread process of rumors terminates and v_1, v_2, v_3, v_7 are still *False* nodes.

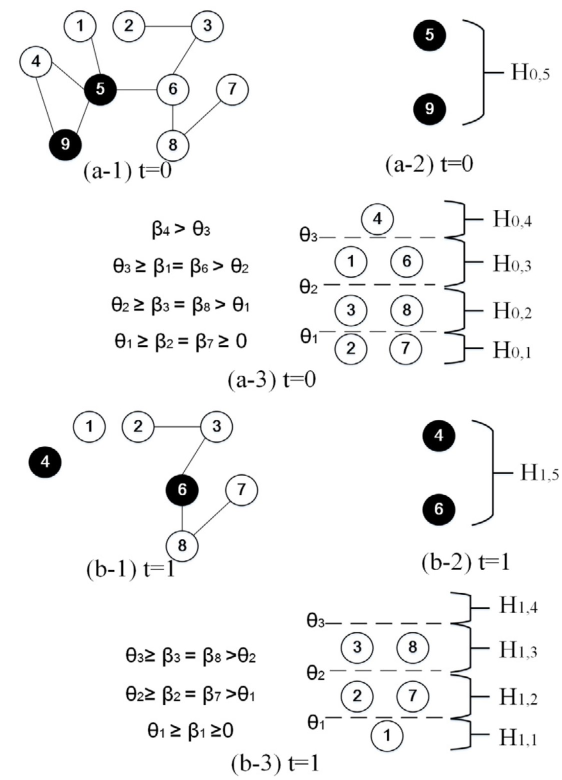

Fig. 2. Illustration of classifying boundaries $(\theta_1, \theta_2, \theta_3)$.

In order to classify different users and quantify the influence weight for each node, we propose a contact coefficient. The *contact coefficient* of node v_j at time t is defined as follows:

$$\beta_{t,j} = \omega_1 x + \omega_2 y + \omega_3 z + \omega_4 \lambda, \tag{3}$$

where $\omega_1, \omega_2, \omega_3, \omega_4$ are the given weight coefficient, x denotes the number of *True* nodes in node v_j's 1st-hop, y denotes the number of *True* nodes in node v_j's 2nd-hop and z denotes the number of *True* nodes in node v_j's 3rd-hop. For the 4th or more hops, λ is defined as:

$$\lambda = \begin{cases} 0, no\ True\ node\ in\ it's\ 4th\ or\ more\ hops \\ 1, have\ True\ nodes\ in\ it's\ 4th\ or\ more\ hops \end{cases} \tag{4}$$

Specially, if a node v_j is an acnode at time t, then $\beta_{t,j} = 0$.

For each node v_j, its contact coefficient $\beta_{t,j} \in [0, +\infty)$, based on which users can be classified into five groups. Let $\theta_1, \theta_2, \theta_3 (0 \leq \theta_1 < \theta_2 < \theta_3)$ denote the dividing boundaries. We aim to find θ_1, θ_2, and θ_3 to classify the users. The nodes can be classified by the following rules:

- H_1: the nodes with contact coefficient in $[0, \theta_1]$.
- H_2: the nodes with a contact coefficient of $(\theta_1, \theta_2]$.
- H_3: the nodes with a contact coefficient of $(\theta_2, \theta_3]$.
- H_4: the nodes with a contact coefficient of $(\theta_3, +\infty)$.
- H_5: the *True* nodes.

For ease of understanding, we suppose there is a social network with 9 nodes, as shown in Fig. 2. In this figure, the black nodes denote the *True* nodes and white nodes denote the *False* nodes. Suppose that $\omega_1 = 2$, $\omega_2 = 1$, $\omega_3 = 0.5$, $\omega_4 = 0.1$, $\theta_1 = 0.9$, $\theta_2 = 1.9$, $\theta_3 = 3.5$.

In Fig. 2(a-1) and (a-2), at time $t = 0$, we can see that v_5, v_9 are *True* nodes and they should be put into set H_5. Then $H_{0,5} = \{v_5, v_9\}$, while others are *False* nodes and we need to calculate their contact coefficients in the spreading process. In Fig. 2(a-3), we figure out that $\beta_{0,4} = 4$, $\beta_{0,1} = \beta_{0,2} = 3$, $\beta_{0,3} = \beta_{0,8} = 1.5$ and $\beta_{0,2} = \beta_{0,7} = 0.6$. Then $\beta_{0,4} > \theta_3$, $\theta_3 \geq \beta_{0,1} = \beta_{0,2} > \theta_2$, $\theta_2 \geq \beta_{0,3} = \beta_{0,8} > \theta_1$ and

$\theta_1 \geq \beta_{0,7} = \beta_{0,2} \geq 0$. Therefore, at time $t = 0$, $H_{0,1} = \{v_2, v_7\}$, $H_{0,2} = \{v_3, v_8\}$, $H_{0,3} = \{v_1, v_6\}$ and $H_{0,4} = \{v_4\}$. As can be seen Fig. 2(b-1) and (b-2), at time $t = 1$, we can see that v_4, v_6 are *True* nodes and they should be put into set H_5. Then $H_{1,5} = \{v_4, v_6\}$, and we figure out that $\beta_{1,1} = 0$, $\beta_{1,2} = \beta_{1,7} = 1$ and $\beta_{1,3} = \beta_{1,8} = 2$. Fig. 2(b-3) shows that $\theta_3 \geq \beta_{1,3} = \beta_{1,8} > \theta_2$, $\theta_2 \geq \beta_{1,2} = \beta_{1,7} > \theta_1$ and $\theta_1 \geq \beta_{1,1} \geq 0$. Therefore, at time $t = 1$, $H_{1,2} = \{v_2, v_7\}$, $H_{1,3} = \{v_3, v_8\}$ and $H_{1,1} = \{v_1\}$, $H_{1,4} = \emptyset$.

3.3. Problem definition

Fast Controlling Rumor (FCR) problem. Given a graph $G = (V, E)$ with n nodes, k *True* nodes, influence probability $p_{(u,v)}$ and the blocking influence rates $(\alpha_1, \alpha_2, \alpha_3, \alpha_4)$, different costs $(C_1, C_2, C_3, C_4, C_5)$, we try to find classifying boundaries θ_1, θ_2 and θ_3, such that τ is minimized within budget C_{total} when the spread is over.

Minimize: τ
Subject to:

$$H_{\theta_1\theta_2\theta_3(cost)} \leq C_{total} \tag{5}$$

Eq. (5) means that the actual cost should not exceed C_{total}.

The actual cost $H_{\theta_1\theta_2\theta_3(cost)}$ is calculated in Eq. (6), which is the sum of the costs of taking different measures from time 0 to time t. Here $|H_{t,i}|$ is the number of nodes in group i at time t and can be formulated as following Eqs. (7)–(10).

$$\begin{aligned} H_{\theta_1\theta_2\theta_3(cost)} &= C_1 \cdot |H_{0,1}| + C_2 \cdot |H_{0,2}| + C_3 \cdot |H_{0,3}| + C_4 \cdot |H_{0,4}| \\ &\quad + C_5 \cdot |H_{0,5}| \\ &\quad + C_1 \cdot |H_{1,1}| + C_2 \cdot |H_{1,2}| + C_3 \cdot |H_{1,3}| + C_4 \cdot |H_{1,4}| \\ &\quad + C_5 \cdot |H_{1,5}| \\ &\quad + \cdots \\ &\quad + C_1 \cdot |H_{t,1}| + C_2 \cdot |H_{t,2}| + C_3 \cdot |H_{t,3}| + C_4 \cdot |H_{t,4}| \\ &\quad + C_5 \cdot |H_{t,5}| \\ &= \sum_{s=0}^{t} \sum_{i=1}^{5} C_i \cdot |H_{s,i}| \end{aligned} \tag{6}$$

where $0 \leq C_1 < C_2 < C_3 < C_4 < C_5$.

$$|H_{t,1}| = \sum_{j=1}^{S_t} \gamma_{t,j}, where \;\; \gamma_{t,j} = \begin{cases} 1, 0 \leq \beta_{t,j} \leq \theta_1 \\ 0, \beta_{t,j} > \theta_1 \end{cases} \tag{7}$$

$$|H_{t,2}| = \sum_{j=1}^{S_t} \eta_{t,j}, where \;\; \eta_{t,j} = \begin{cases} 0, \beta_{t,j} \leq \theta_1 \\ 1, \theta_1 < \beta_{t,j} \leq \theta_2 \\ 0, \beta_{t,j} > \theta_2 \end{cases} \tag{8}$$

$$|H_{t,3}| = \sum_{j=1}^{S_t} \xi_{t,j}, where \;\; \xi_{t,j} = \begin{cases} 0, \beta_{t,j} \leq \theta_2 \\ 1, b < \beta_{t,j} \leq \theta_3 \\ 0, \beta_{t,j} > \theta_3 \end{cases} \tag{9}$$

$$|H_{t,4}| = \sum_{j=1}^{S_t} \mu_{t,j}, where \;\; \mu_{t,j} = \begin{cases} 0, \beta_{t,j} \leq \theta_3 \\ 1, \beta_{t,j} > \theta_3 \end{cases} \tag{10}$$

where S_t denotes the number of *False* nodes at time t.

4. Proposed method

In this section, we design a Withdraw Bedeckung-Greedy Algorithm (WB-GA) for finding the shortest τ in social networks. In proposed WB-GA algorithm, let $h_i (i = 1, 2, \ldots,)$ denote the number of nodes with the same contact coefficient, as shown in Algorithm 1.

In Algorithm 1, we find the $\theta_1, \theta_2, \theta_3$ based on contact coefficient. We rank nodes with the same contact coefficient in descending order and put the *True* nodes into H_5. Case 1, we randomly select different $\theta_1, \theta_2, \theta_3$ to satisfy serveral conditions. If $H_{\theta_1\theta_2\theta_3(cost)} \leq C_{total}$, output $\theta_1, \theta_2, \theta_3$. Case 2, under the cost constraint, the nodes with the same contact coefficient are together successively put into a set from H_4, H_3, H_2 to H_1. Then if the cost is less than C_{total}, the algorithm will move the nodes with the smallest contact coefficient in H_4 to H_3 and move the nodes with the largest contact coefficient in H_1 to H_2 until get the better solution. Finally, the largest contact coefficients of nodes H_1, H_2 and H_3 are θ_1, θ_2 and θ_3, respectively.

It should be noted that every time when we judge whether $H_{\theta_1\theta_2\theta_3(cost)} \leq C_{total}$, we have to recalculate the $H_{\theta_1\theta_2\theta_3(cost)}$. Algorithm 2 shows that we sum up all the costs from the beginning of the spread until all rumors terminate calculate $H_{\theta_1\theta_2\theta_3(cost)}$.

Algorithm 1: (WB-GA). Input: A graph $G = (V, E)$, rumor seed set R, C_{total}, C_1, C_2, C_3, C_4, C_5, $p_{(u,v)}$, α_1, α_2, α_3, α_4, ω_1, ω_2, ω_3, ω_4. Output: $\theta_1, \theta_2, \theta_3$, τ.

```
Initialization: θ1 = θ2 = θ3 = 0, |H5| = |R|, n' = n − |R|
Delete the nodes where v_j^sta = True and calculate the β_{t,j}
Sort the different values in descending order(β1 > β2 > ...βm) and
    count h_m corresponding to each β_m.Let a = 10^4 · des · |R|/n,(des is
    the density of G). Select the different values of θ1, θ2, θ3 randomly
    to satisfy ((if a ≤ 0.02, then|H4| > n'/200, |H3| > 0.01n'),
    (elself 0.02 < a ≤ 0.1, then|H4| > 3n'/200, |H3| > 0.03n'),
    (elseif 0.1 < a ≤ 0.5, then|H4| > n'/3, |H3| > 0.05n'),
    (else, |H4| > n'/2, |H3| > 0.04n'), |H2| > 0, |H1| > 0).
if H_{θ1θ2θ3(cost)} ≤ C_total, (/*case 1*/) then
    Output the value of θ1, θ2, θ3, τ
else
    (/*case 2*/)
    for h_p(p ∈ {1, 2, 3, ..., m}) do
        Add h_p nodes to the H4 and determine whether
        H_{θ1θ2θ3(cost)} ≤ C_total is satisfied. If yes, p = p + 1. Otherwise,
        add h_p nodes to the H3, H2 in turn and determine whether
        H_{θ1θ2θ3(cost)} ≤ C_total is satisfied. Until h_p can only be added to
        H1 or p = m, find the value of θ1, θ2, θ3, τ then break
    end for
    According to θ1, θ2, θ3, calculate H1 and H4
    if H1 ≠ ∅ and H4 ≠ ∅ then
        Let h_r and h_w be equal to the number of nodes with the
        largest contact coefficient in H3 and H1, respectively.
    end if
    Let p = r, q = w + 1, h_0 = h_1
    while h_{p+1} · (p_(u,v) − α3) < h_p · (p_(u,v) − α4), p ∈ {r, r − 1, ..., 1} and
    H1 ≠ ∅ and H4 ≠ ∅ do
        Remove h_{p+1} nodes from H4 and add them to H3
        for q ≤ m and H_{θ1θ2θ3(cost)} ≤ C_total do
            Remove h_q nodes from H1 and add them to H2
            q = q − 1
        end for
        p = p + 1
    end while
    Output the value of θ1, θ2, θ3, τ
end if
```

Algorithm 2: (CALCULATE $H_{\theta_1\theta_2\theta_3(cost)}$) Input: C_1, C_2, C_3, C_4, C_5, α_1, α_2, α_3, α_4, $\theta_1, \theta_2, \theta_3$, k. Output: $H_{\theta_1\theta_2\theta_3(cost)}$, τ.

```
Initialize t = 0, u = k, H_{θ1θ2θ3(cost)} = 0, S_t = n − k, C' = 0.
for The expected number of new True nodes is not less than 1 do
    Calculate every node's contact coefficient (β_{t,j}(j ∈ {1, 2, 3, ...S_t}))
    Calculate the |H_{t,1}|, |H_{t,2}|, |H_{t,3}|, |H_{t,4}| based on the θ1, θ2, θ3 and
    C' = Σ_{i=1}^{5} C_i · |H_{t,i}|. Delete the nodes where v_j^sta = True
    Let S_t = S_t − |H_{t,5}| and H_{θ1θ2θ3(cost)} = C' + H_{θ1θ2θ3(cost)}
    t = t + 1
end for
Output τ = t, H_{θ1θ2θ3(cost)}
```

Now, we illustrate the method of calculating $H_{\theta_1\theta_2\theta_3(cost)}$ in Fig. 1. We still suppose that $\alpha_1 = 0.1$, $\alpha_2 = 0.2$, $\alpha_3 = 0.3$, $\alpha_4 = 0.4$. $\omega_1 = 2$, $\omega_2 = 1$, $\omega_3 = 0.5$, $\omega_4 = 0.1$, $\theta_1 = 0.9$, $\theta_2 = 1.9$, $\theta_3 = 3.5$. Assume that the costs of different controlling measures are C_1, C_2, C_3, C_4 and C_5.

First, according to Fig. 1(a), at time $t = 0$, $cost_0 = C_1 \cdot |H_{0,1}| + C_2 \cdot |H_{0,2}| + C_3 \cdot |H_{0,3}| + C_4 \cdot |H_{0,4}| + C_5 \cdot |H_{0,5}| = 2C_1 + 2C_2 + 2C_3 + C_4 + 2C_5$. As shown in Fig. 1(b), we delete the *True* nodes. Then we calculate that $cost_1 = C_1 \cdot |H_{1,1}| + C_2 \cdot |H_{1,2}| + C_3 \cdot |H_{1,3}| + C_5 \cdot |H_{1,5}| = C_1 + 2C_2 + 2C_3 + 2C_5$. Similarly, in Fig. 1(c), $cost_2 = C_1 \cdot |H_{2,1}| + C_3 \cdot |H_{2,3}| + C_5 \cdot |H_{2,5}| = 3C_1 + C_3 + C_5$. In Fig. 2(d), at time $t = 3$, there are no more new *True* nodes and the spread is over. Therefore, $cost_3 = 0$. Finally, add up all the costs, then $H_{\theta_1\theta_2\theta_3(cost)} = cost_0 + cost_1 + cost_2 = 6C_1 + 4C_2 + 5C_3 + 2C_4 + 5C_5$.

5. Theoretical analysis

In this section, we conduct theoretical analysis on the properties of FCR problem and algorithm WB-GA.

Theorem 1. *Let f_t be the expected number of True nodes when the spread is over at time t, then calculating the f_t is #P-hard.*

Proof. As shown in [36] that $s - t$ connectedness problem is #P-complete. Based on [37], this problem is equivalent to calculating the probability of connecting two nodes when each edge in G is connected with a probability of $1/2$.

Then we reduce this problem to calculating f_t as follows. Let $f_t(G)$ denote the expected number of nodes which are not be influenced at time t in G and S_0 denote the number of nodes which are not be influenced at time $t = 0$ in G. Without loss of generality, we suppose that $p_e = 0.5$ for all edges $e \in E$. Then we add nodes $u_j (j = \{1, 2, 3...\})$ and the corresponding directed edges from u_j to w_j to the graph G to make up the new graph G'. Let $p_{(U,G)}$ denote the probability U under S_0 in G. We can indicate that $f_t(G') = f_t(G) + p_{(u_j,w_j)} \cdot p_{(U,G)}$, where $U = \{u_j | j = 1, 2, 3...\}$. Therefore, let $p_{(u_j,w_j)} = 1$, the probability that any node $v \in S_0$ connects to node u_1 in G' is $p_{(U,G)}$. Thus we converted the problem of calculating f_t to the $s - t$ connectedness problem and it is #P-hard. □

Corollary 2. *FCR problem is NP-hard.*

Lemma 3. *For FCR problem, if there have a viable solution under the C_{total}, algorithm WB-GA can find $\theta_1, \theta_2, \theta_3$ in the finite-time period.*

Proof. In algorithm WB-GA case 1, it is obviously true. In case 2, the worst case happens when each node has a different contact coefficient. Then the algorithm will put each node into a set from H_4, H_3, H_2 to H_1. The algorithm will terminate when all nodes are put into different H_i under the cost constraint. We prove by contradiction that this terminal condition must be satisfied in finite time. Assume that when all nodes are put into different H_i, the cost is larger than C_{total}. According to the WB-GA, more nodes will be put into the H_1. If all nodes are put into H_1, then there must be a solution and the cost must be less than C_{total}. This contradicts our assumption. □

Lemma 4. *According to the algorithm WB-GA, the τ of the FCR problem is finite.*

Proof. In this paper, given the network, we just consider that each node can be influenced by rumors at most once [24,30–32]. The worst case happens when there is only one new rumor node at each time. According to the algorithm WB-GA, if the node is influenced by rumors, it would be deleted from the set V. Therefore, the τ is finite and $\tau \leq n$. □

The core-components of algorithm WB-GA is to find the $\theta_1, \theta_2, \theta_3$ based on contact coefficient and dynamically divide nodes at different moments into different groups. The algorithm ranks nodes with the same contact coefficient in descending order. In case 1, we randomly select different $\theta_1, \theta_2, \theta_3$ to satisfy serveral conditions. If $H_{\theta_1\theta_2\theta_3(cost)} \leq C_{total}$, output $\theta_1, \theta_2, \theta_3$. In case 2, under the cost constraint, the nodes with the same contact coefficient are together successively put into a set from H_4, H_3, H_2 to H_1. Then if the cost is less than C_{total}, the algorithm will dynamically move the node with the smallest contact coefficient in H_4 and the node with the largest contact coefficient in H_1 until get the better solution. Finally, the largest contact coefficients of nodes H_1, H_2 and H_3 are θ_1, θ_2 and θ_3, respectively.

Because the *FCR* problem is *NP*-hard, the optimal solution cannot be found in polynomial time. Therefore, we try to use algorithm WB-GA to find the approximate solution. Moreover, our algorithm has low time complexity and can get results with a data-parameter-based parameter-based approximation ratio on different datasets.

Theorem 5. *The time complexity of the proposed algorithm WB-GA is $O(m + n)$ (not include Algorithm 2), where n is the number of nodes of the network and m is the number of edges of the network.*

Proof. In the case 2 of WB-GA, nodes with different contact coefficients will be put into different H_i. The worst case happens when in each round, each node has a different contact coefficient, which means that the node is put into the set one by one. Then, the algorithm removes nodes from H_4 and add them to H_3 and at the same time, the algorithm removes nodes from H_1 and add them to H_2. The worst case happens when there are x nodes need to be removed from H_4 one by one and $(n - x - 2)$ nodes need to be removed from H_1 one by one. Thus, the time complexity is $O(n)$. Besides, in case 1 or 2, we need to calculate the contact coefficient based on the degree of each node. Thus, the time complexity is $O(m)$. Therefore, the overall running time of WB-GA is $O(m + n)$. □

Theorem 6. *The algorithm WB-GA is correct.*

Proof. For Algorithm 1 (WB-GA) case 1, it is obviously true. For case 2, the proof process is divided into the following two parts.

Part I

At the initial time, $H_4 = H_3 = H_2 = H_1 = \emptyset$. For the first iteration of the loop ($p = 1$), the nodes with the largest contact coefficients will be added to H_4. If the $H_{\theta_1\theta_2\theta_3(cost)}$ is larger than C_{total}, these nodes will be added to H_3 or H_2 or H_1 until $H_{\theta_1\theta_2\theta_3(cost)} \leq C_{total}$. It can be indicated that the loop invariant ($p = 1$) holds for the first iteration of the loop.

For the next iterations of the for loop, each time, the nodes with the largest contact coefficients will also be added to H_4 or H_3 or H_2 or H_1 under the condition that $H_{\theta_1\theta_2\theta_3(cost)} \leq C_{total}$. Then each iteration can always maintain this invariant($p = 2, 3, \ldots, m$).

Finally, when the for loop terminates with $p > m$, for each iteration of the loop, p increases by 1, then it must have the case of that p is equal to $m + 1$. If we replace $p = m$ with $p = m + 1$ in the circular invariant, we cannot add any node to H_4 or H_3 or H_2 or H_1. Then, we can infer that there are already all nodes in H_4 or H_3 or H_2 or H_1. Therefore, this part is correct.

Part II

For the first iteration of the loop, algorithm will remove the nodes from H_4 and add them to H_3. Then algorithm will remove nodes from H_1 and add them to H_2 until $H_{\theta_1\theta_2\theta_3(cost)}$ is largest . It can be indicated that the loop invariant ($H_{\theta_1\theta_2\theta_3(cost)} \leq C_{total}$) holds for the first iteration of the loop.

For the next iterations of the for loop, each time, the algorithm will still remove the nodes from H_4 and add them to H_3 and remove nodes from H_1 and add them to H_2. Then each iteration can always maintain this invariant($H_{\theta_1\theta_2\theta_3(cost)} \leq C_{total}$).

Finally, when the for loop terminates with $H_{\theta_1\theta_2\theta_3(cost)} > C_{total}$. Therefore, this part is correct. □

Theorem 7. *If $(\alpha_4-\alpha_3) << (\alpha_2-\alpha_1)$ and $p_{(u,v)} << 1$, then in case 2, the data-parameter-based approximation ratio of the proposed algorithm WB-GA is more than $\log_{(1-p_{(u,v)}+\alpha_3)}(1-p_{(u,v)}+\alpha_2)$, where $p_{(u,v)}$ is the influence probability for each edge without taking any measure, α_2 is the blocking influence rate of H_2 and α_3 is the blocking influence rate of H_3.*

Proof. Let *opt* be the optimal solution of the *FCR* problem. For convenience, write τ^* for *opt* and τ' be the result of the algorithm WB-GA. In the case of τ^*, let r_1, r_2, r_3, r_4 be the proportions of H_1, H_2, H_3, H_4 in the total *False* nodes, respectively. In the case of τ', let $\mu_1, \mu_2, \mu_3, \mu_4$ be the proportions of H_1, H_2, H_3, H_4 in the total *False* nodes, respectively.

For convenience, let

$$\Phi_1 = p_{(u,v)} - \alpha_1,\ \Phi_2 = p_{(u,v)} - \alpha_2$$
$$\Phi_3 = p_{(u,v)} - \alpha_3,\ \Phi_4 = p_{(u,v)} - \alpha_4$$
$$A = 1-(r_1+r_2)\Phi_2-(r_3+r_4)\Phi_3$$
$$B = 1-r_1\Phi_1-r_2\Phi_2-r_3\Phi_3-r_4\Phi_4$$

Then, we can conclude that

$$\frac{\tau'}{\tau^*} \geq \log_{(1-r_1\Phi_1-r_2\Phi_2-r_3\Phi_3-r_4\Phi_4)}(A) = \log_{(B)}(A)$$

where $r_1+r_2+r_3+r_4=1$

We define that

$$F(r_1,r_2,r_3,r_4,\zeta) = \frac{-ln(A)}{-ln(B)} + \zeta(r_1+r_2+r_3+r_4-1)$$

Then applying the method of Lagrange multiplier, we have

$$F'_{(r_1)} = \frac{-\Phi_2(A)^{-1}ln(B)}{-ln^2(1-r_1\Phi_1-r_2\Phi_2-r_3\Phi_3-r_4\Phi_4)} - \frac{-\Phi_1(B)^{-1}ln(A)}{-ln^2(1-r_1\Phi_1-r_2\Phi_2-r_3\Phi_3-r_4\Phi_4)} + \zeta = 0$$

$$F'_{(r_2)} = \frac{-\Phi_2(A)^{-1}ln(B)}{-ln^2(1-r_1\Phi_1-r_2\Phi_2-r_3\Phi_3-r_4\Phi_4)} - \frac{-\Phi_2(B)^{-1}ln(A)}{-ln^2(1-r_1\Phi_1-r_2\Phi_2-r_3\Phi_3-r_4\Phi_4)} + \zeta = 0$$

$$F'_{(r_3)} = \frac{-\Phi_3(A)^{-1}ln(B)}{-ln^2(1-r_1\Phi_1-r_2\Phi_2-r_3\Phi_3-r_4\Phi_4)} - \frac{-\Phi_3(B)^{-1}ln(A)}{-ln^2(1-r_1\Phi_1-r_2\Phi_2-r_3\Phi_3-r_4\Phi_4)} + \zeta = 0$$

$$F'_{(r_4)} = \frac{-\Phi_3(A)^{-1}ln(B)}{-ln^2(1-r_1\Phi_1-r_2\Phi_2-r_3\Phi_3-r_4\Phi_4)} - \frac{-\Phi_4(B)^{-1}ln(A)}{-ln^2(1-r_1\Phi_1-r_2\Phi_2-r_3\Phi_3-r_4\Phi_4)} + \zeta = 0$$

$$\Phi_1 F'_{(r_2)} - \Phi_2 F'_{(r_1)} = \frac{(A)^{-1}}{-ln(B)} \times \Phi_2(\Phi_2-\Phi_1) - \zeta(\Phi_2-\Phi_1) = 0$$

$$\Phi_3 F'_{(r_4)} - \Phi_4 F'_{(r_3)} = \frac{(A)^{-1}}{-ln(B)} \times \Phi_3(\Phi_4-\Phi_3) - \zeta(\Phi_4-\Phi_3) = 0$$

$$\zeta = \frac{\Phi_2(1-(r_1+r_2)\Phi_2-(r_3+r_4)\Phi_3)^{-1}}{-ln(1-r_1\Phi_1-r_2\Phi_2-r_3\Phi_3-r_4\Phi_4)} = \frac{\Phi_3(1-(r_1+r_2)\Phi_2-(r_3+r_4)\Phi_3)^{-1}}{-ln(1-r_1\Phi_1-r_2\Phi_2-r_3\Phi_3-r_4\Phi_4)}$$

We can infer that iff $\alpha_2=\alpha_3$, the function ζ holds. Therefore, we just calculate the lower bound. Let $f(A) = -ln(A)$ and we can know that $0 < A \leq 1$ and $f(A)$ is a monotonically decreasing function. Then we have

$$ln(A) \geq ln(1-(r_1+r_2)\Phi_2-(r_3+r_4)\Phi_2) = ln(1-\Phi_2)$$

Therefore

$$\frac{\tau'}{\tau^*} > \frac{-ln(1-\Phi_2)}{-ln(1-\Phi_3)} = \log_{(1-\Phi_3)}(1-\Phi_2) = \log_{(1-p_{(u,v)}+\alpha_3)}(1-p_{(u,v)}+\alpha_2)$$

where $0<\alpha_2<\alpha_3<1$. □

Table 2
Social network datasets.

Network	Nodes	Edges	Type
Lastfm-social	7624	27,806	Social network of LastFM from Asia
Deezer-social	28,281	92,752	Social network of Deezer from Europe

Lemma 8. *Let τ_n be the spread time of rumors without any measures and let τ_y be the spread time of rumors with at least one control measure.*

Then in the *FCR* problem, we can have $\tau_y < \tau_n$ or $\tau_y = \tau_n$ or $\tau_y > \tau_n$.

Proof. Suppose that node u is a *True* node and other nodes are *False* nodes and $p_{(u,v)} = 1$. Then we divide possibilities into the following two cases:

Case 1. There is only one path that node u can influence v_m

In Fig. 3(a), we assume that $\tau_n = m$ and $v_1, v_2, v_3, \dots, v_m$ are *False* nodes. Without losing generality, we assume that we take measures to the node $v_j (1 \leq j \leq m)$. After taking measures to node v_j, node v_{j-1} cannot influence v_j and the spread process is over. Then we can infer that the $\tau_y = j-1 \leq m-1 < \tau_n$.

Case 2. There are at least two path that node u can influence v_m

We assume that $\tau_n = m$ and $v_1, v_2, v_3, \dots, v_m$ and $w_1, w_2, w_3, \dots, w_r$ are *False* nodes, where $r < m$. Then we have the following two subcases:

Subcase 2.1.

In Fig. 3(b), we assume that we take measures to the node w_r. Then we can infer that $\tau_y = m = \tau_n$.

Subcase 2.2.

In Fig. 3(c), we assume that we take measures to the node $w_j (1 < j < r)$. When node v_m is influenced by v_{m-1}, v_m can still continue to influence w_r. Then we can infer that $\tau_y \geq m+1 > \tau_n$. □

6. Experiment

6.1. Datasets and parameters

We select two real social networks from SNAP[1] and use them to evaluate the effectiveness of our algorithm.

(1) Lastfm-social: It is an online social network of LastFM users, which is collected from the public API in March 2020. Nodes are LastFM users from Asian countries and edges are relationships between them. Its density is 0.001.

(2) Deezer-social: It is an online social network of Deezer users, which is collected from the public API in March 2020. Nodes are Deezer users from European countries and edges are relationships between them. Its density is 0.0002.

The details of these datasets are shown in Table 2.

The probability on edges can be uniformly set as $p_{(u,v)} = 0.05$ or $p_{(u,v)} = 0.1$ [38] or other values. Moreover, we set $\omega_1 = 2$, $\omega_2 = 1$, $\omega_3 = 0.5$ and $\omega_4 = 0.1$. Then we set different values for k (the number of initial rumor seed nodes) and C_{total} (partial) as shown in Table 3.

For other parameters, we randomly set different combinations of C_i, α_i and the details are shown in Table 4.

[1] https://snap.stanford.edu/.

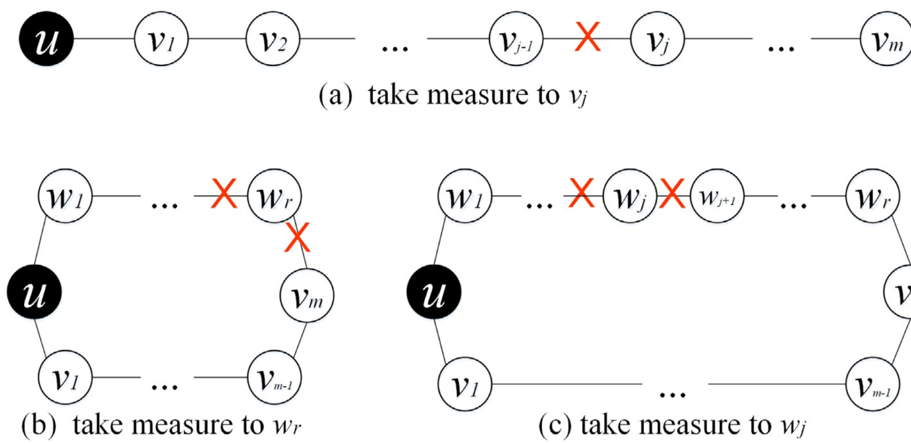

Fig. 3. Illustration of the spread time of rumors.

Table 3
Parameter values (partial)

Parameter	Value
k	50, 100, 150, 200, 400, 600, 800
C_{total}	4×10^3, 7×10^3, 9×10^3, 1.1×10^4, 1.3×10^4, 1.8×10^4, 2×10^4, 2.2×10^4, 2.6×10^4, 3×10^4, 1.2×10^6, 1.7×10^6, 2.1×10^6, 2.4×10^6, 3×10^6

Table 4
Different combinations of C_i and α_i.

Combination	Parameter	$i=1$	$i=2$	$i=3$	$i=4$	$i=5$
1	C_i	0	5	20	50	150
	α_i	0%	1.5%	2.5%	3.5%	–
2	C_i	0	2	4	8	16
	α_i	0%	1.5%	2.5%	3.5%	–
3	C_i	0	1	10	100	1000
	α_i	0%	7%	8%	9%	–
4	C_i	0	2	4	8	16
	α_i	0%	2%	4%	8%	–

Table 5
f_r with different k under Lastfm-social when $p_{(u,v)}=0.05$ and $Comb=1$.

Algorithm	$k=50$	100	150	200	400	600	800
$Unblocking$	274.2	392.4	458.5	503.7	615.2	682.4	731.2
$Proximity$	267.3	389.1	455.9	502.0	613.0	680.2	728.8
$IMRank$	262.3	380.3	446.0	492.8	605.1	672.9	722.0
$Degree$	244.6	372.7	441.0	488.8	602.1	670.8	720.0
$WB-GA$	14.0	27.8	43.9	77.8	137.0	173.0	217.3

Table 6
f_r with different k under Deezer-social when $p_{(u,v)}=0.05$ and $Comb=2$.

Algorithm	$k=50$	100	150	200	400	600	800
$Unblocking$	135.2	236.9	327.1	401.3	629.0	810.4	957.2
$Proximity$	129.6	232.5	324.0	398.7	627.4	808.2	955.4
$IMRank$	130.7	229.5	318.2	392.7	619.4	798.6	946.1
$Degree$	111.6	212.9	303.8	380.1	612.1	792.5	941.5
$WB-GA$	14.8	57.2	112.4	196.2	324.5	462.9	567.0

6.2. Comparison methods

For a fair comparison, for each algorithm, we use 1000 Monte-Carlo to estimate the results of the expected spread time of rumors (τ) and the expected number of nodes influenced by rumors (denoted as f_r, except the initial rumors). In addition, in real life, various measures can be taken only if the cost is large enough. Therefore, we set C_{total} to be large enough when we do the experiment of case 1. The experiment of this case is mainly used to show the superiority of our strategy of classifying nodes based on the contact coefficient. In the case 2 of algorithm, we need to use Monte Carlo simulation to calculate $H_{\theta_1\theta_2\theta_3(cost)}$ when we adjust the nodes with the same contact coefficient, which needs huge amounts of running time. Therefore, we do not show this case in the experiment, but we make a theoretical analysis of this case in section 5. In addition, for all comparison methods (except $Unblocking$), we set the number of H_4, H_3, H_2 to all be 10. Our algorithm does not need to set the number of each H_i, because our algorithm classifies all nodes at each time based on the contact coefficient. In our experiment, all algorithms are written in C++ and running on an Intel(R) Core(TM) i5-9500 with 3.30 GHz CPU and 32 GB RAM. Then we choose four methods to select nodes and take different measures. The four methods for comparison as follows:

(i) Proximity [24]. It is a popular heuristic algorithm which selects and classifies the neighbors with the highest index of the rumor seed nodes. Specially, we give an index to each node.

(ii) Degree [35]. It selects and classifies nodes with the largest degree.

(iii) IMRank [39]. It selects and classifies nodes with the interplay between calculation of ranking-based marginal influence spread and ranking.

(iv) Unblocking [40]. This is a special case of taking no action.

6.3. Experimental results

For convenience, in the following figures, we use $Comb = 1$ to denote the *Combination 1* of C_i and α_i in Table 4 that $\alpha_1 = 0\%$, $\alpha_2 = 1.5\%$, $\alpha_3 = 2.5\%$, $\alpha_4 = 3.5\%$ and $C_1 = 0$, $C_2 = 5$, $C_3 = 20$, $C_4 = 50$, $C_5 = 150$. Similarly, we use $Comb = 2$, $Comb = 3$ and $Comb = 4$ to denote the *Combination 2*, *Combination 3* and *Combination 4* of Table 4, respectively.

Fig. 4 show the expected number of *True* nodes (f_r) and τ of different algorithms under Lastfm-social when $p_{(u,v)} = 0.05$ and $Comb = 1$.

In Fig. 4(a), we can see that f_r of all algorithms increases with the k. Moreover, the f_r of algorithm WB-GA is between 14 nodes and 217 nodes and it is less than that of the other algorithms when the k is the same. Because we set the number of each H_i to be small, algorithm *Unblocking* and algorithms *Proximity*, *IMRank*, *Degree* have the similar f_r.

Furthermore, Table 5 shows the f_r of all algorithms under Lastfm-social in more detail. We can clearly see that f_r of algorithm *Unblocking* is larger than f_r of other algorithms in Table 5. The f_r of algorithm WB-GA is less than that of the other algorithms when the k is the same.

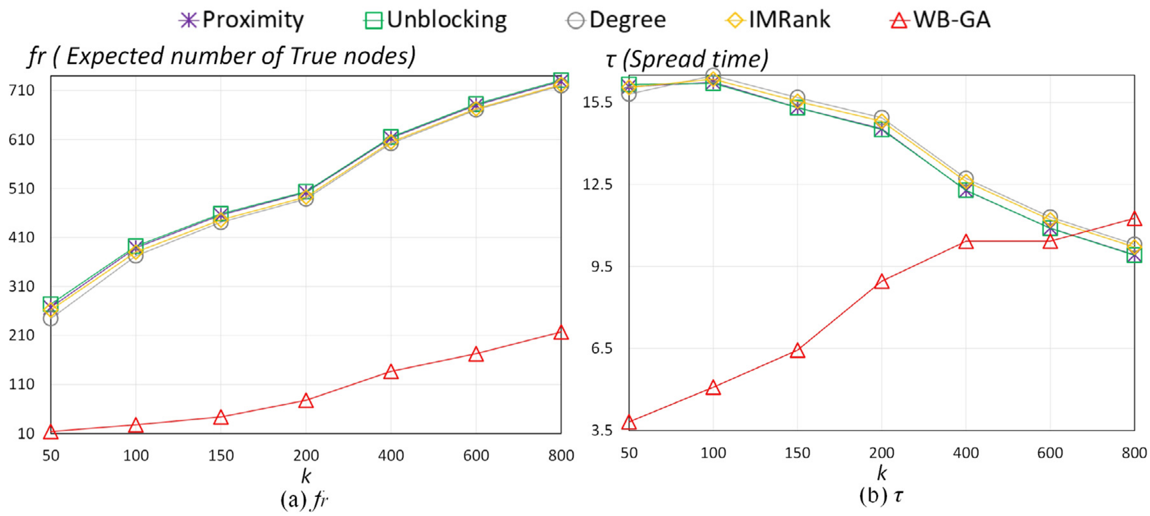

Fig. 4. The f_r and τ under Lastfm-social when $p_{(u,v)} = 0.05$ and $Comb = 1$.

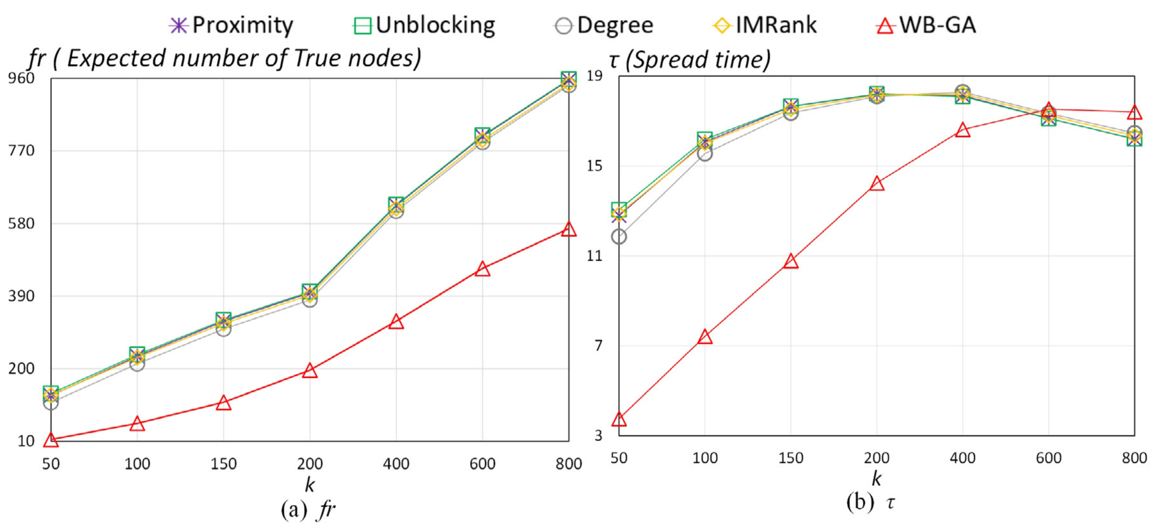

Fig. 5. The f_r and τ with different k under Deezer-social when $p_{(u,v)} = 0.05$ and $Comb = 2$.

In Fig. 4(b), we can also see that the τ of WB-GA is less than that of the other algorithms when the $k < 800$. When $k = 800$, it can be inferred that WB-GA is not effective because there are too many initial rumor nodes and in practice, k is usually small. The τ of WB-GA is roughly between 4 and 11 when the $p_{(u,v)} = 0.05$ under dataset Lastfm-social. The τ of other algorithms decreases with the k when $k \geq 50$. We can infer that the smaller the k, the better the performance of our algorithm. In addition, we can see that the τ of algorithm *Unblocking* is slightly less than that of the *Degree* and *IMRank* and the τ of algorithm *Unblocking* is the same as that of algorithm *Proximity*. It means that although we have taken measures, the spread time may also increase or remain the same. In the meantime, this also confirms Lemma 8. In a word, WB-GA outperforms the other algorithms on both f_r and τ under Lastfm-social when $k < 800$, $p_{(u,v)} = 0.05$ and $Comb = 1$.

Table 6 shows the f_r of different algorithms under Deezer-social when $p_{(u,v)} = 0.05$ and $Comb = 2$. It also shows the similar trends as Table 5. Compared to Table 5, we can see that f_r of algorithm WB-GA is much less than f_r of WB-GA in Table 6 when $k > 100$. That is because the number of nodes of dataset Deezer-social is more than that of dataset Lastfm-social.

Fig. 5 shows the f_r and τ of different algorithms under Deezer-social when $p_{(u,v)} = 0.05$ and $Comb = 2$. In Fig. 5(a) and (b), we can know that the τ of our algorithm WB-GA is less than that of the other algorithms when the $k < 600$. The τ of WB-GA is roughly between 4 and 17 under dataset Deezer-social. Fig. 5 also shows similar trends as Fig. 4. However, comparing Fig. 5 and Fig. 4(b), the τ of WB-GA is larger in Fig. 5. That may be because the average degree or density of dataset Deezer-social is less than the average degree of dataset Lastfm-social and the number of nodes of Deezer-social is larger. Meanwhile, our algorithm WB-GA also performs better than the other algorithms under Deezer-social when $p_{(u,v)} = 0.05$ and $k < 600$.

Fig. 6 shows the classifying boundaries of WB-GA under two datasets when $p_{(u,v)} = 0.05$. Fig. 6(a), (b) and (c) show the results of θ_1, θ_2 and θ_3, respectively. We can see that $\theta_1, \theta_2, \theta_3$ increase with the k under dataset Lastfm-social *(Comb = 1)* or under dataset Deezer-social *(Comb = 2)*. Because the number nodes of dataset Deezer-social is larger than that of dataset Lastfm-social, the values of $\theta_1, \theta_2, \theta_3$ under Deezer-social are larger than that of the Lastfm-social.

In Fig. 7, the cost of all algorithms is less than or equal to the given C_{total}. Fig. 7(a) shows the actual cost of algorithms under Lastfm-social when $p_{(u,v)} = 0.05$ and $Comb = 1$. Fig. 7(b) shows the actual cost of algorithms under Deezer-social when $p_{(u,v)} = 0.05$ and $Comb = 2$.

From Fig. 7(a) and (b), we can see that the actual cost of our algorithm WB-GA is more than that of the other algorithms when the k is the same. That is because we set the number of H_i of other algorithms is small. Theoretically, we can also get a smaller cost of our algorithm

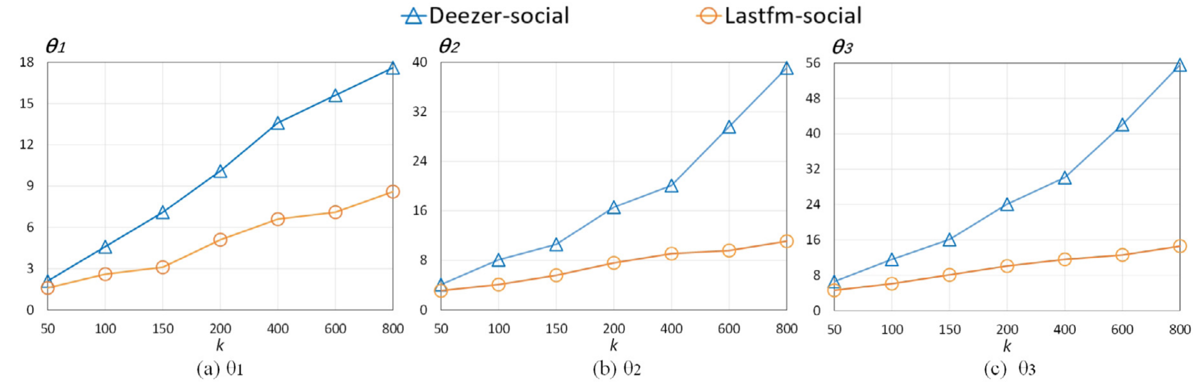

Fig. 6. Values of $\theta_1, \theta_2, \theta_3$ of WB-GA under different datasets when $p_{(u,v)} = 0.05$.

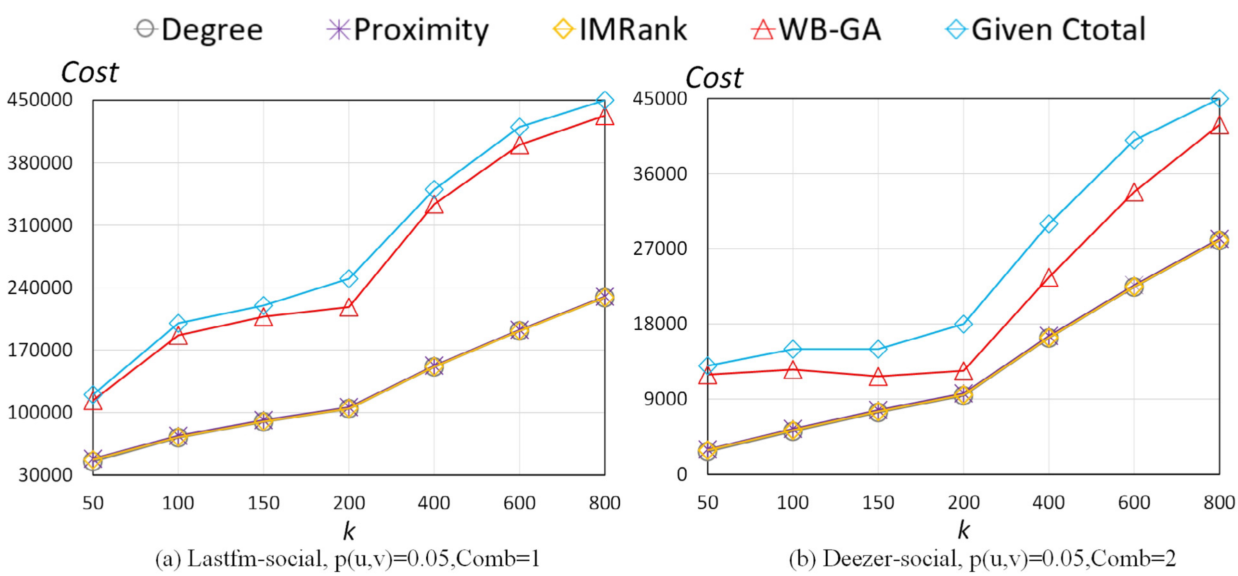

Fig. 7. The cost of different algorithms under two datasets with the given C_{total}.

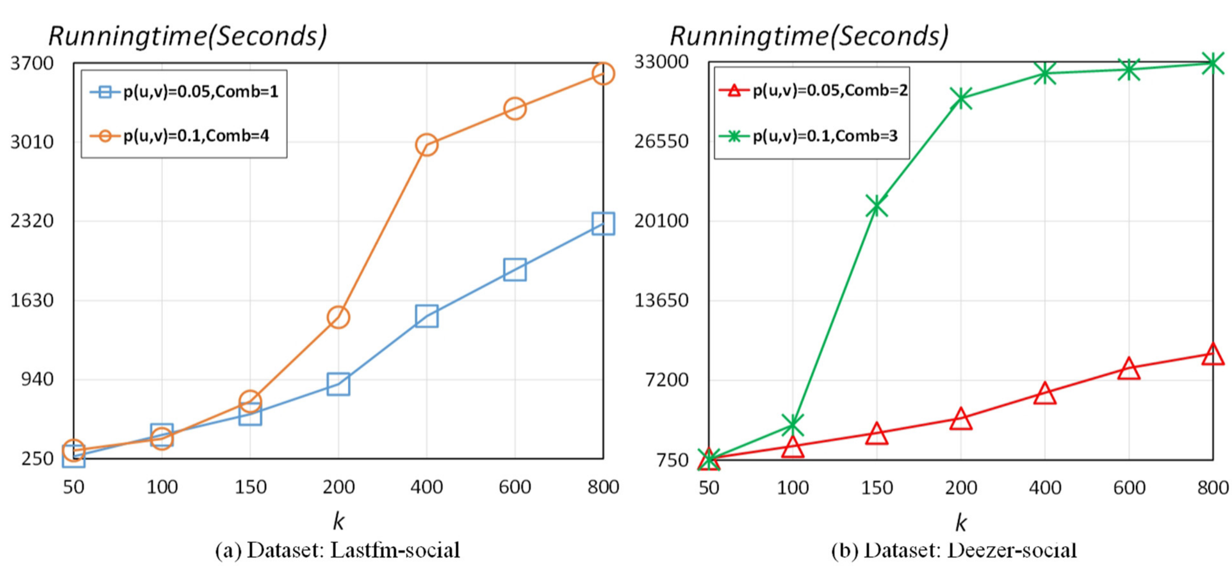

Fig. 8. Running time of WB-GA under two datasets with different $p_{(u,v)}$ and *Comb*.

through the case 2 of WB-GA. We can also see that the actual cost of WB-GA under Lastfm-social is larger than that of WB-GA under Deezer-social. That may be because the cost in $Comb = 1$ of dataset Lastfm-social is larger than the cost in $Comb = 2$ of dataset Deezer-social. Specially, *Unblocking* is taking no action, therefore we do not show the cost of the algorithm *Unblocking* here.

Fig. 8 shows the running time of WB-GA under different dataset with different $p_{(u,v)}$ and it includes the running time of 1000 Monte Carlo simulations. In Fig. 8(a), we can see that the actual running time of WB-GA is roughly between 4.5 min and 38 min when $p_{(u,v)} = 0.05$ and from 5 min and 60 min when $p_{(u,v)} = 0.1$ under the dataset Lastfm-social. In Fig. 8(b), we can see that the actual running time of WB-GA is roughly between 15 min and 156 min when $p_{(u,v)} = 0.05$ and from 13 min and 548 min when $p_{(u,v)} = 0.1$ under the dataset Deezer-social. It can also be seen that the running time of WB-GA increases as the k. In addition, there is not much difference in running time between $p_{(u,v)} = 0.05$ and $p_{(u,v)} = 0.1$ when $k < 150$. Moreover, the running time of dataset Deezer-social is more than that of Lastfm-social. This is because the number of nodes in dataset Deezer-social is larger than that in dataset Lastfm-social.

7. Conclusions

In this paper, we study the problem of fast controlling of rumors (FCR) problem in social networks. First, we propose five measures for different users to control rumors, including *taking no action*, *tagging the user*, *blocking access to information*, *spreading the truth* and *deleting the user's account*. Taking five measures to control rumors is more realistic and effective than taking only one. Second, based on the IC model, we propose a MPIC model to describe the process of spreading rumors. This model has multiple influence probabilities, which is of more practical significance than a single influence probability. Then in order to classify users, we propose a contact coefficient which can intuitively quantify the influence weight for each user in social network. More importantly,based on it, users can be dynamically divided into different groups at each moment. We discuss that the *FCR* problem is *NP*-hard, and calculating the rumor nodes is $\#P$-hard. Then we design an algorithm WB-GA based on contact coefficient, which can not only control the spread of rumor in the shortest time with limited cost, but also ensure that the data-parameter-based approximate ratio of the result is more than $\log_{(1-p_{(u,v)}+\alpha_3)}(1-p_{(u,v)}+\alpha_2)$. Finally, the simulation of real datasets shows that our algorithm WB-GA can effectively find the boundaries of different users and the performances of our algorithm are superior to that of the comparison methods.

CRediT authorship contribution statement

Xiaopeng Yao: Investigation, Conceptualization, Methodology, Design algorithm, Prove theories, Writing draft. **Yue Gu:** Generate some figures, Design algorithm and programming, Put forward some corollaries. **Chonglin Gu:** Algorithm improvement, Optimization of time complexity and algorithm, Revise the paper. **Hejiao Huang:** Put forward ideas, Discuss and optimize algorithm and method, Revise the paper.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is financially supported by Shenzhen Science and Technology Program under Grant No. JCYJ20210324132406016 and National Natural Science Foundation of China under Grant No. 61732022, China Postdoctoral Science Foundation under Grant No. 2020T130633, Guangdong Basic and Applied Basic Research Foundation, China under Grant No. 2019A1515110214 and Initial Scientific Research Fund of HITSZ, China with Grant No. BB45001011.

References

[1] N. Difonzo, N.M. Robinson, J.M. Suls, C. Rini, Rumors about cancer: Content, sources, coping, transmission, and belief, J. Health Commun. 17 (9) (2012) 1099–1115.
[2] H.J. Oh, H. Lee, When do people verify and share health rumors on social media? The effects of message importance, health anxiety, and health literacy, J. Health Commun. 24 (11) (2019) 837–847.
[3] E. Khalil, B. Dilkina, L. Song, Scalable diffusion-aware optimization of network topology, in: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Datamining (2014) pp. 1226–1235.
[4] Kimuram. Saittok, . Motodah, Minimizing the Spread of Contamination By Blocking Links in a Network, Vol. 8, AAAI, 2008, pp. 1175–1180.
[5] Y. Ping, Z. Cao, H. Zhu, Sybil-aware least cost rumor blocking in social networks, in: Proc. Globecom, IEEE, 2014, pp. 692–697.
[6] M. Kimura, K. Saito, H. Motoda, Blocking links to minimize contamination spread in a social network, ACM Trans. Knowl. Discov. Data 3 (2) (2009) 1–23.
[7] H. Habiba, Y. Yu, T.Y. Berger-Wolf, J. Saia, Finding spread blockers in dynamic networks, in: Proc. SNAKDD (2010) pp. 55–76.
[8] S. Wang, X. Zhao, Y. Chen, Z. Li, J. Xia, Negative influence minimizing by blocking nodes in social networks, in: Proc. 27th, AAAI Conference on Artificial Intelligence (2013) pp. 134–136.
[9] L. Yang, Z.W. Li, A. Giua, Containment of rumor spread in complex social networks, Inf. Sci. (2020) 113–130.
[10] Z. Tan, D. Wu, T. Gao, I. You, V. Sharma, Aim: Activation increment minimization strategy for preventing bad information diffusion in OSNs, Future Gener. Comput. Syst. (2018) 293–301.
[11] A.I.E. Hosni, K. Li, S. Ahmad, Minimizing rumor influence in multiplex online social networks based on human individual and social behaviors, Inf. Sci. (2020) 1458–1480.
[12] B. Wang, G. Chen, L. Fu, L. Song, X. Wang, DRIMUX: DYnamic rumor influence minimization with user experience in social networks, IEEE Trans. Knowl. Data Eng. 29 (10) (2017) 2168–2181.
[13] G. Tong, W. Wu, L. Guo, D. Li, C. Liu, B. Liu, D. Du, An efficient randomized algorithm for rumor blocking in online social networks, IEEE Trans. Netw. Sci. Eng. 7 (2) (2020) 845–854.
[14] Y. Liu, X. Chen, Y. Rao, H. Xie, Q. Li, J. Zhang and, et al., Supervised group embedding for rumor detection in social media, in: Proc. 19th, ICWE, 2019, Vol. PP, pp. 139-153.
[15] T. Takahashi, N. Igata, Rumor detection on Twitter, in: Proc. 6th Int. Conf. Soft Comput. Intell. Syst and 13th Int. Symp. Adv. Intell. Syst. (2012) pp. 452–457.
[16] Z. Wang, W. Dong, W. Zhang, C.W. Tan, Rumor source detection with multiple observations: Fundamental limits and algorithms, in: Proc. ACM Int. Conf. Meas. Model. Comput. Syst. (2014) pp. 1–13.
[17] V. Qazvinian, E. Rosengren, D.R. Radev, Q. Mei, Rumor has it: Identifying misinformation in microblogs, in: Proc. Conf. Empirical Methods Natural Language Process. (2011) pp. 1589–1599.
[18] S. Kwon, M. Cha, K. Jung, W. Chen, Y. Wang, Prominent features of rumor propagation in online social media, in: Proc. ICDM (2013) pp. 1103–1108.
[19] J. Ma, W. Gao, Z. Wei, Y. Lu, K. Wong, Detect rumors using time series of social context information on microblogging websites, in: Proc. CIKM (2015) pp. 1751–1754.
[20] F. Yu, Q. Liu, S. Wu, L. Wang, T. and Tan, A convolutional approach for misinformation identification, in: Proc. IJCAI (2017) pp. 3901–3907.
[21] J. Ma and, et al., Detecting rumors from microblogs with recurrent neural networks, in: Proc. IJCAI, 2016, pp. 3818-3824.
[22] Z. He, Z. Cai, J. Yu, X. Wang, Y. Sun, Y. Li, Cost-efficient strategies for restraining rumor spreading in mobile social networks, IEEE Trans. Veh. Technol. (2017) 2789–2800.
[23] R. Yan, D. Li, W. Wu, D. Du, Y. Wang, Minimizing influence of rumors by blockers on social networks: Algorithms and analysis, IEEE Trans. Netw. Sci. Eng. (2020) 1067–1078.
[24] G. Tong, W. Wu, L. Guo, D. Li, C. Liu, B. Liu, D. Du, An efficient randomized algorithm for rumor blocking in online social networks, in: Proc. Annu. Joint Conf. IEEE Computer Communications (2017) pp. 1–9.
[25] H. Jing, H. Liang, Y. Hao, Controlling infection by blocking nodes and links simultaneously, in: Proc. WINE (2011) pp. 206–217.
[26] X. Liu, S. Li, X. Liao, S. Peng, L. Wang, Z. Kong, Know by a handful the whole sack: Efficient sampling for top-k influential user identification in large graphs, World Wide Web-Internet Web Inf. Syst. 17 (4) (2014) 627–647.
[27] H. Zhang, H. Zhang, X. Li, M.T. Thai, Limiting the spread of misinformation while effectively raising awareness in social networks, in: Prof. Int. Conf. Comput. Social Netw. (2015) pp. 35–47.
[28] C.J. Kuhlman, G. Tuli, S. Swarup, M.V. Marathe, S. Ravi, Blocking simple and complex contagion by edge removal, in: Proc. of the 13th Data Mining (ICDM), International Conference on, IEEE, 2013, pp. 399–408.

[29] P. Dey, S. Roy, Centrality based information blocking and influence minimization in online social network, in: Proc. IEEE International Conference on Advanced Networks and Telecommunication Systems (ANTS), IEEE, 2017, pp. 1–6.

[30] C. Budak, D. Agrawal, A. El Abbadi, Limiting the spread of misinformation in social networks, in: Proc. of the 20th WWW, ACM, 2011, pp. 665–674.

[31] X. He, G. Song, W. Chen, Q. Jiang, Influence blocking maximization in social networks under the competitive linear threshold model, in: SDM, SIAM, 2012, pp. 463–474.

[32] L. Fan, Z. Lu, W. Wu, B. Thuraisingham, H. Ma, Y. Bi, Least cost rumor blocking in social networks, in: Proc. of the 33rd ICDCS, IEEE, 2013, pp. 540–549.

[33] K. Scaman, A. Kalogeratos, N. Vayatis, A greedy approach for dynamic control of diffusion processes in networks, in: Proc. of the 27th International Conference on Tools with Artificial Intelligence (ICTAI), IEEE, 2015, pp. 652–659.

[34] A. Kalogeratos, K. Scaman, N. Vayatis, Learning to suppress sis epidemics in networks, in: Proc. Networks in the Social and Information Sciences workshop, 2015.

[35] D. Kempe, J. Kleinberg, E. Tardos, Maximizing the spread of influence through a social network, in: Proc. 9th ACM SIGKDD Int. Conf. Knowledge. Discovery Data Mining (2003) pp. 137–146.

[36] L.G. Valiant, The complexity of enumeration and reliability problems, SIAM J. Comput. 8 (3) (1979) 410–421.

[37] W. Chen, C. Wang, Y. Wang, Scalable influence maximization for prevalent viral marketing in large-scale social networks, in: Proc. 16th SIGKDD Int. Conf. Knowledge. Discovery Data Mining, 2010, pp. 1029–1038.

[38] Y. Zhang, W. Yang, D. Du, Rumor correction maximization problem in social networks, Theoret. Comput. Sci. (2021) 102–116.

[39] S. Cheng, H. Shen, J. Huang, W. Chen, X. Cheng, IMRank: Influence maximization via finding self-consistent ranking, in: Proc. of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval, (2014) pp. 475–484.

[40] Q. Fang, X. Chen, Q. Nong, Z. Zhang, Y. Cao, Y. Fen, and, et al., General rumor blocking: An efficient random algorithm with martingale approach, Theoret. Comput. Sci. (2020) 82–93.