Knowledge-Based Systems 231 (2021) 107451

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Node deletion-based algorithm for blocking maximizing on negative influence from uncertain sources

Weijia Ju [a], Ling Chen [a,*], Bin Li [a], Yixin Chen [b], Xiaobing Sun [a]

[a] College of Information Engineering of Yangzhou University, Yangzhou 225127, China
[b] Department of Computer Science, Washington University in St. Louis, USA

ARTICLE INFO

Article history:
Received 30 January 2021
Received in revised form 5 July 2021
Accepted 25 August 2021
Available online 30 August 2021

Keywords:
Linear threshold model
Social network
Negative influence blocking

ABSTRACT

The spreading of negative influence, such as epidemic, rumor, false information and computer virus, may lead to serious consequences in social networks. The issue of negative influence blocking maximization arouses intense interest of the researchers. However, in the real world social network environment, the exact source of negative influence is usually unknown. In most cases, we only know the distribution of negative seeds, which is the probability for each node to be a negative seed. In this work, we investigate the problem of maximizing the blocking on negative influence from uncertain sources. We propose the competitive influence linear threshold propagation model (CI-LTPM) for the problem. Based on the IC-LTPM model, we define the problem of uncertain negative source influence blocking maximization (UNS-IBM). We use the propagation tree in the live-edge (LE) sub-graph for estimating the influence propagation. An algorithm is proposed to calculate the blocking increments of the positive seeds based on the propagation tree in the LE sub-graph. We observed that the blocking effect of the positive seeds is the reduction on the negative influence after the positive seeds and their related edges being deleted from the LE sub-graph. Based on such observation, we propose a node deletion-based algorithm *NDB* (node-deletion-blocking) for solving the UNS-IBM problem. Our experiment results show that *NDB* can block more negative influence in less computational time than other methods.

1. Introduction

In recent years, social network is constantly expanding, more and more people are participating in activities on social networks. With the fast expansion of social networks, network analysis has provided a powerful tool for the researches in many fields such like link prediction [1], information diffusion [2–4], topological analyzing [5], community partitioning [6], direction revival [7], metric learning [8], and epidemic control [9]. Social network also provides an ideal site for the businesses to advertise their commodity. Usually, a merchant may promote his new product by choosing some influential customers to propagate the positive information of its product to a large scope. An important issue in this marketing strategy is how to select such influential customers to persuade a maximum number of customers to accept the product. This is the problem of influence maximization (IM) [10,11], which has aroused intense interest of the researchers. The purpose of IM is to identify some influential nodes which can spread the influence to the maximal scope in the social network. We refer to such influential nodes as the seeds of influence spreading.

While the population of the users in the networks increases rapidly, various viewpoints are posted in the network by the users, and they might possess positive or negative opinions on the same topic. Such different views reflect the competitions between the businesses or the political organizations. As a result, positive and negative influences may spread in the social network simultaneously [6,8]. It is desirable to spread the positive influence to the maximum range of individuals in the network, and to limit the negative one of the competitors [12].

Due to the openness and high speed of the social networks, all kinds of false information and rumors could be propagated very fast and widely among the users [13,14]. Such false messages are usually accompanied by network hot spots or eye-catching information, and tend to attract wide attention of the users in the network. The false news will cause harm to the individuals in the news to some extent, and may lead to social panic. A rumor could decline the credibility of a business, and may damage the reputation of a person. Furthermore, the spread of fake news may lead to a serious loss of social network's reputation and security. Therefore, such misinformation must be controlled to make social network more reliable and secure in information exchanging. We

* Corresponding author.
E-mail address: lchen@yzu.edu.cn (L. Chen).

https://doi.org/10.1016/j.knosys.2021.107451
0950-7051/

must find efficient approaches to limit the destructive impact of negative influence. To effectively block the spreading of negative influence on social networks, a common used way is to choose the influential individuals who can propagate the true news to minimize the influence of false news [15]. If users receive true information and correct the false one, they will not be affected by the harmful information any more. Therefore, the spread of positive information can block the propagation of negative one in social networks. The critical concern in this problem is how to detect the persuasive individuals who can spread the positive information to maximally counteract the influence of the negative one. This is the problem of negative influence blocking maximization (IBM) [16]. Different from the IM problem, which detects the seed set to spread the positive influence to the maximum scope in the network, IBM problem detects the positive seed set to propagate positive information to maximally block the negative influence.

Although the IBM problem has been extensively investigated [17], and many effective methods have been reported [18–21], there are still some challenges. The existing works are based on the assumption that the negative seed set, which is the source of negative information, is always known when we detect the positive seed set to counteract the negative one. However, in the applications on real world social network environment, it is difficult to obtain the exact sources of the negative influence. Due to the privacy rule in the social network, it is hard to identify the sources of the rumor and misinformation. In many cases, we only know the distribution of the negative seeds, namely, the probability for each node to be a negative seed. Such distribution of the negative influence source is used to describe the uncertainty of the negative seeds, and may be estimated by the empirical knowledge or the observation on the behaviors of the users. For example, in the spread of an epidemic, we do not know the original source of the epidemic, but we can approximately know its distribution from the information of the infected people. Those who have the closest contact with the infected people may have higher probability to be the sources of infection. For another example, in the spread of rumors, we do not know the source of the rumor, but we can get its approximate distribution based on the information of the rumor receivers. The influential person who frequently communicates with the receivers may have higher probability to be the source of the rumor. In the transmission of a computer virus, we can estimate the distribution of the virus source using information of the infected computers. The network nodes which have exchanged more information with the infected computers will have higher probability to be the source. In these cases, it is a challenge to detect the positive seed set which maximizes the blocking on the negative influence from the uncertain sources.

In this study, we define this problem as uncertain negative source influence blocking maximization (UNS-IBM) under the linear threshold (LT) propagation model. Due to the uncertainness of the negative influence sources, the UNS-IBM problem is quite different from the traditional IBM problem. The main challenges of the UNS-IBM problem are:

1. The traditional IBM problem detects the positive seed set based on a determined negative seed set. We can estimate the blocking effect of a positive seed set using a simulation on the influence spreading under the joint impacts of positive and negative seed sets. But in the UNS-IBM problem, the negative seed set is unknown, it is difficult to evaluate the blocking effect of a positive seed set by simulation or other approaches.

2. To solve the traditional IBM problem, we need to test only one negative seed set which provides a unique base for detecting the positive seed set. But in the UNS-IBM problem, it consumes huge amount of computation time to enumerate and test all the $2^{|V|}$ possible negative seed sets.

Since UNS-IBM is much more difficult than the traditional IBM problem, the methods for IBM are no longer applicable for the UNS-IBM problem. Motivated by this background, we investigate the UNS-IBM problem in this study. Our goal is to devise an effective algorithm to detect the optimal positive seed set which can maximize the blocking effect on the negative influence from uncertain source. The main contributions and innovations of this work include:

1. We define an extended LT propagation model CI-LTPM for the competitive social networks where the positive and negative influences are spreading simultaneously. Based on the CI-LTPM model, we formally define the problem of uncertain negative source influence blocking maximization (UNS-IBM).

2. To avoid the time-consuming simulation on influence spreading process, we propose the propagation tree in the LE sub-graph for estimating the influence propagation.

3. An algorithm is proposed to calculate the blocking increments of the positive seeds based on the propagation trees in the LE sub-graph.

4. We propose a node deletion-based algorithm *NDB* (node-deletion-blocking) for solving the UNS-IBM problem.

Based on the above mentioned contributions, the key results of our work are:

1. We prove that the objective function of the node deleting-based method is monotonical and sub-modular, which guarantees the effectiveness of the method.

2. We propose a method to estimate the number of LE sub-graphs so that error of the results can be restricted within a given bound ε under a probability ρ. Experimental results show that based on the estimated number of LE sub-graphs, our method can achieve well trade-off between the computation time and the accuracy of the results.

3. Abundant experimental results show that the proposed node deletion-based method can block more negative influence than the other methods. In addition, the computational time required is less than the other methods.

The remainder of this paper is organized as follows. Section 2 gives a review on related works. Section 3 extends the LT propagation model to the competitive social networks, and formally defines the UNS-IBM problem. Section 4 proposes an algorithm to estimate the block increment of the positive seed node based on the propagation tree in the LE sub-graph. Section 5 presents a node deletion-based algorithm *NDB* to maximize the blocking on the negative influence from the uncertain sources. Section 6 shows and analyzes the experimental results. Section 7 discusses the results of our work. Section 8 concludes the paper and indicates our future work.

2. Related works

In recent years, many influence spreading models and methods for influence maximization (IM) have been proposed [10, 11] by extending the traditional linear threshold (LT) and independent cascade (IC) propagation models in networks. Şimsek et al. [22] changed the structure of the IM problem in order to tailor it to swarm intelligence algorithms. They sorted the nodes based on their influential levels. Then, the nodes can be mapped into a latent space so that nodes with similar influence level are placed at the nearby locations in the space. They employed swarm intelligence approach to obtain the optimal solution of the IM problem. To obtain a larger diffusion effect, Li et al. [23] took two parts of the nodes into consideration. One part consists of "elites" which are the influential nodes, while the other part consists of "grassroots" which are the ordinary nodes. They tried to choose the seeds from some common grassroots, and proved that they are more suitable to be selected as seeds than the elites in influence maximization. They developed an algorithm to select grassroots as seeds based on community partition. CaliÒet al. [24] investigated the issue of targeted influence maximization which maximizes the influence on a given group of target users. They developed a method which ensures an ε(1-1/e-ε)-approximate solution for the issue. Li et al. [25] investigated the targeted influence maximization in a multifactor based influence spreading model, and presented a heuristic algorithm to maximize the influence spreading on the target users. Yang et al. [26] presented a double-layer influence spreading model and a seed selection method to maximize the influence spreading. Chen et al. [2] investigated the problem of semantics-aware influence maximization. They presented a method for the problem based on the extended reverse influence set.

In some competitive social networks, two types of influences may spread simultaneously: positive and negative influences. We must maximize the spread of the positive influence, and minimize the negative influence of the rival. Recently, works on positive influence maximization have been reported [6,8,12,27–29]. Wang et al. [29] defined this issue as maximizing the positive influenced users (MPIU), and used the fluid dynamics theory to describe the dynamic influence spreading. They proposed a greedy-based approach to identify the positive seeds. Singh et al. [27] studied the problem of *maximizing multi-type influences in the multiple networks. They* assumed that each node can be activated by multiple influences. They also proved that this problem is NP-hard, and its objective function is sub-modular in both IC and LT spreading models. They presented a heuristic approach to identify the seed set which can maximize the *multi-type influence* spreading in multiple networks. Based on the simulated annealing method, Li et al. [8] presented an algorithm for maximizing the positive influence in the networks. In addition, they proposed heuristic methods to obtain higher quality results. Yang et al. [30] studied the issue of relative influence maximization (RIM) and presented effective methods to obtain the largest relative influence, i.e. to propagate positive influence and reduce the spreading of the negative one. Khomami et al. [31] proposed a learning automata-based approach to detect the influential nodes such that the positive influence can be propagated to the maximum range. Based on community structure of the network, Bozorgi et al. [32] presented an approach for the problem of positive influence maximization under a modified LT model.

In recent years, studies have been conducted on the problem of negative influence blocking maximization (IBM). Influence spreading models for the problem of influence maximization are extended for the IBM problem. By extending the independent cascade spreading model, Budak et al. [33] presented a multi-campaign IC model (MCICM) and defined the eventual influence limitation (EIL) problem. They gave the sufficient conditions for the objective function of EIL being sub-modular. They also presented an approximate algorithm for the EIL problem under the MCICM model. Liu et al. [16] extended the classical LT model for influence spreading and suppression, and presented methods for the IBM problem under this model. Deng et al. [34] studied the influence maximization in heterogeneous networks. They proposed a measuring influence (MIF) model to capture social influence with heterogeneity. They proposed a greedy-based algorithm to solve the IBM problem. The algorithm selects the nodes with the maximum marginal influence as the seeds.

Recently, various methods for the IBM problem are reported to detect the positive seed users for blocking the negative influence spreading. Budak et al. [33] studied the issue of negative influence contamination under the multi-campaign IC model. They presented a method to identify a set of positive seed users to minimize the spreading of the negative information. Lv et al. [18] proposed an algorithm named CB-IBM for IBM in the campaign-oblivious independent cascade model. Shi et al. [19] defined the problem of adaptive influence blocking (AIB) which detects the positive seeds adaptively according to the dynamic spread of the negative influence. They proposed a method called k-nodes-per-round to identify the positive seeds based on an α-Tolerance selecting policy. Jankowski [35] studied how the suppressing action on negative influence is related to the action time and the campaign intensity. They showed that the delay in suppressing process can be compensated by choosing proper parameters. Arazkhani et al. [36] presented a community-based algorithm FC_IBM to minimize the diffusion of negative information. They used fuzzy clustering and centrality measures to find the positive seed set.

In some methods for negative influence blocking, greedy approach is employed to detect the positive seeds. Wang et al. [37] proposed a greedy-based method to minimize the spreading of the negative influence by blocking some uninfluenced users. Also based on the greedy seed selection strategy, Yao et al. [38] proposed a method for minimizing the negative influence by blocking a limited number of links in a network. Zhang et al. [39] proposed a method to limit the misinformation from known sources. They used the greedy approach and applied the CELF heuristic to identify the positive seeds. Based on the linear threshold model with one direction state transition (LT1DT), Yang et al. [40] proposed a method named ProxMinGreedy for negative influence containment in social networks. The method uses the greedy approach to minimize the influence of the false information.

Some methods employ heuristic optimization to find the positive seed set for the IBM problem. Based on node eigenvector centrality, Yao et al. [20] presented a heuristic method to minimize the negative influence spreading by choosing a set of positive seeds. Zhu et al. [21] investigated the location-based IBM problem, which attempts to maximally block the negative influence spreading in a given geographical area by a set of positive seeds. Based on the quad-tree data structure, they presented two heuristic algorithms for the problem. To solve the IBM problem in the competitive IC spreading model, Wu et al. [17] proposed two heuristic approaches named CMIA-H and CMIA-O based on the largest influence propagation tree. Archbold et al. [41] investigated the problem of minimizing the spread of a target concept based on a model which describes the interactions between the spreading concepts. They also presented a heuristic method for limiting concept spread.

Considering that the behavior of influence propagating and blocking is essentially a game between the two competitive parts, Tsai et al. [42] modeled the opposite strategies of influence spreading and blocking as a zero-sum game by treating the selection of the positive and negative seed sets as the strategic space. In order to avoid enumerating the huge strategy space, they proposed a game Nash equilibrium-based algorithm to select the positive seeds using the strategy generating technology. Li et al. [43] investigated the issue of negative influence blocking with strategic propagation sources. They modeled the influence blocking as a min–max optimization problem, and tempted to minimize the negative influence propagation range.

An important application of IBM is rumor controlling in the social networks [13]. In recent years, rumor control has attracted much attention of the researches in public opinion monitoring and political competition. The goal of rumor control is in two aspects: one is to detect the source of rumor in the network, the other is to prevent the rumor spreading by propagating positive news. To identify the rumor sources, many rumor spreading models are proposed in recent years. *Jiang et al. [44] proposed a two stage model to simulate the real situation of news break-out. They proposed a method to detect the positive public opinion disseminators by considering the probability of rumor spreaders and the variation of rumor influenced rate.* Huo et al. [45] presented a rumor propagation model and studied the stability of the equilibrium between rumor-free and rumor-endemic. The model provides a reasonable description for the rumor transmission. Some researchers expended and improved the epidemic model SIR (susceptible–infected–recovered) by considering the topological structure of the social network. Chen [46] presented a rumor transmission model by taking the cooling off period of the transmitters and the motility of the individuals in a given range into consideration. Stability of the model was analyzed and the strategy of rumor controlling in emergency was proposed. Dong et al. [47] presented a double-identity rumor spreading model by considering the identities of the rumor creator, spreaders, controllers and the common individuals. Different features of the identities are considered to ensure the diversity of these identities. A forgetting mechanism was presented based on the impact of the interactions between the identities.

Recently, various methods for rumor controlling have been reported. Yang et al. [5] presented a diffusion dynamics-based approach for minimizing the rumor spreading. Wang et al. [7] investigated the issue of rumor monitoring by considering multiple surveys. He et al. [48] studied the rumor controlling problem in mobile network. Hosni et al. [49] presented a rumor propagation model HISB (human individual and social behaviors). Under the HISB model, they proposed a greedy method to limit the rumor spreading in social networks. They [50] also proposed a method to select the positive seed nodes based on likelihood maximization. The method can get a (1-1/e)-approximation result. Zhu et al. [51] proposed a rumor spreading model based on a silence-forcing function in networks. Under this model, an optimal control method was proposed to reduce the range of rumor transmission. Guo et al. [52] studied the issue of targeted protection maximization (TPM) which is to protect the targeted users from being influenced by the rumor. They proposed two methods to solve the TPM problem.

In a real world social network, it is difficult to identify the exact source of the negative influence due to the privacy rule in the network. Usually the negative seed set is unknown in advance, and we only know its distribution which indicates the probability for each node to be a negative seed. The distribution can be estimated by the empirical knowledge or the observation of the behaviors of the competitive users. Such distribution of the influence source causes the uncertainty of the negative seeds. The key issue is how to detect a set of positive seeds which can maximally block the negative influence from the uncertain sources. This is the problem of uncertain negative source influence blocking maximization (UNS-IBM). The problem is challenging since it is difficult to calculate the blocking effect of a positive seed set on the uncertain negative seeds by simulation or other approaches. In addition, it requires large amount of computational time to enumerate and test all the possible negative seed sets. Therefore, the methods for the traditional IBM are not capable for solving the UNS-IBM problem. In this study, we investigate the UNS-IBM problem, and present a node deletion-based method for detecting the optimal positive seed set which can maximize the expected blocking effect on the negative influence from uncertain sources.

3. Model the influence propagation for IBM and UNS-IBM

In this Section, we first define the problem of uncertain negative source influence blocking maximization (UNS-IBM). By extending the LT propagation model, we propose an approximate model IC-LTPM for the UNS-IBM problem. Based on the live-edge (LE) sub-graph under the IC-LTPM model, the influence spreading can be estimated. We present an algorithm to construct the LE sub-graph. Finally, we show how to determine a proper sampling size of the LE sub-graphs.

The main notations used in this paper are shown in Table 1.

3.1. Problem formulation

In a competitive social network, positive and negative influences may spread simultaneously. In each moment of influence spreading, a node stays in one of the three states: Before being influenced, it is in the inactive state; once it is positively influenced, it shifts to the positively activated state; if it is negatively influenced, it is switched to the negatively activated state. If an inactive node is positively (negatively) influenced, it will stay in positively (negatively) activated state, and cannot be activated by the negative (positive) influence. In a competitive social network, a merchant or a politician usually chooses some influential individuals to propagate his favorable information on the network to limit the information of its competitors. The spreading of positive information can counteract the effect of the negative influence in social networks [26,46]. The key issue is how to identify the persuasive individuals who can spread the positive information to maximally counteract the negative effects. This is the problem of negative influence blocking maximization (IBM).

Let $G = (V, E)$ be a competitive network, where V is the node set and E is the edge set. Each directed edge (u, v) in E is assigned a real number $P(u, v)$ indicating the probability for node u to influence its neighbor v.

Definition 1 (*Negative Influence Blocking*)**.** Let D and C be the sets of negative and positive seeds respectively, which spread negative and positive influences in the network. Let $\delta(V, D)$ be the number of nodes being negatively influenced by seed set D, and $\delta(V\backslash C, D)$ be the number of the negatively influenced nodes under the joint effect by seed sets D and C. The negative influence blocking by the positive seed set C is the reduction on the number of the nodes negatively activated:

$$B(C, D) = \delta(V, D) - \delta(V\backslash C, D) \tag{1}$$

Definition 2 (*Negative Influence Blocking Maximization (IBM)*)**.** Let D be a set of negative seed nodes in network $G = (V,E)$, k>0 be a constant. Negative influence maximization is to detect a positive seed set C^* of size k such that the negative influence blocking can be maximized:

$$C* = \underset{C\in V\backslash D, |C|\le k}{\arg\max} B(C, D) \tag{2}$$

The IBM problem can be formulated as an optimization for maximizing the block function $B(C, D)$. It has been proven that the objective function of IBM is monotone and sub-modular, and the IBM problem is NP-hard.

However, in real world social network environment, the exact source of negative information is usually unknown, and the negative seed set D is hard to obtain. In most occasions, we only know the distribution of negative seeds. In this case, it is a challenge to detect a positive seed set C which maximizes the expected blocking on the negative influence from the uncertain sources. This is the problem of uncertain negative source influence blocking maximization.

In a network with uncertain negative influence sources, each node v will be assigned a number $P^-(v)$ indicating the probability for node v to be a negative seed. Here, function P^-: $V\rightarrow(0,1)$ is the distribution of negative seeds in the network. The probability for a node set D being chosen as a negative seed set is defined as

$$P^-(D) = \prod_{v\in D} P^-(v) \prod_{v\notin D} [1 - P^-(v)] \tag{3}$$

Definition 3 (*Uncertain Negative Source Influence Blocking Maximization (UNS-IBM)*)**.** Let H be the set of all the possible negative

Table 1
Main notations used in the paper.

Notation	Description
$G = (V, E)$	A network with node set V and edge set E
$G' = (W, E_w)$	A sub-graph of G, $W \subset V$, E_w is a set of edges connecting with the nodes in W
$P(u, v)$	Probability of influence propagation via edge (u, v)
$\Pr(G'\|G)$	Probability of generating sub-graph G' from G
$\delta_{G'}(V, v)$	Number of the nodes in V which v can reach
$p(v, G', G)$	Probability for node v connecting with at least one edge in G'
$T^v_{G'}$	Propagation tree rooted at v in G'
$r(v, T^v_{G'})$	Set of nodes which v can reach in $T^v_{G'}$
$I\left(u, v, G'\right)$	A function indicating the existence of a path from v to u in G'
$I(u, T^v_{G'})$	A function indicating the existence of a path from v to u in $T^v_{G'}$
Δ(x)	Blocking increment of a candidate positive seed x

seed sets under the distribution P^-. Given a constant k>0 and the distribution P^-, negative influence blocking maximization under uncertain sources is to detect a node set $C^* \subset V$ consisting of k positive seeds such that the expectation of the negative influence blocking by C^* is maximized, namely, to detect a positive seed set C^* satisfying:

$$C^* = \underset{C \in V, |C| \le k}{argmax} \sum_{D \subset H} P^-(D) * B(C, D) \tag{4}$$

Let $G(C)$ be the expected blocking effect of positive seed set C. To approximate $G(C)$, we construct a subset of H consisting of N negative seed sets: $H_N = \{D_1, D_2, \ldots, D_N\}$. When N is large enough, we have

$$G(C) = \sum_{D \in H} P^-(D) * B(C, D) \approx \sum_{D \in H_N} P^-(D) * B(C, D)$$

Let the different seed sets sampled in H_N be $D^1_N, D^2_N, \ldots, D^m_N$ ($m \le N$), and the number of D^i_N's occurrences in H_N be n_i. Then the frequency of D^i_N's occurrences in H_N is n_i/N. Using $B(C)$ to denote the expected negative influence blocking on H_N by positive seed set C, we have:

$$B(C) = \sum_{i=1}^{m} \frac{n_i}{N} B\left(C, D^i_N\right) = \frac{1}{N} \sum_{i=1}^{m} n_i B\left(C, D^i_N\right) = \frac{1}{N} \sum_{i=1}^{N} B(C, D_i) = \frac{1}{N} \sum_{i=1}^{N} \sum_{d \in D_i} B(C, d)$$

Here, $B(C, d)$ is the negative influence blocking value of positive seed set C on the negative influence by the seed d. We define the function:

$$In(d, D) = \begin{cases} 1 & \text{if } d \in D \\ 0 & \text{otherwise} \end{cases}$$

Then, we have

$$B(C) = \frac{1}{N} \sum_{i=1}^{N} \sum_{d \in V} B(C, d) * In(d, D) = \sum_{d \in V} B(C, d) \frac{1}{N} \sum_{i=1}^{N} In(d, D) \approx \sum_{d \in V} B(C, d) P^-(d)$$

Therefore, we approximate the problem of (4) by:

$$C^* = \underset{C \in V, |C| \le k}{argmax} B(C) \tag{5}$$
$$\approx \underset{C \in V, |C| \le k}{argmax} \sum\nolimits_{d \in V} P^-(d) * B(C, d)$$

To overcome the difficulty of unknown negative seeds in the UNS-IBM problem, we can use (5) to estimate the blocking on the influence of each possible negative seed. If for all the candidate positive seed sets C, we know their blockings $B(C,d)$, we can choose the set C with the maximum weighted summation of the $B(C,d)$ values as the positive seed set. Therefore, the key issue is how to estimate the blocking value $B(C,d)$ for the candidate positive seed set C and the possible negative seed d. We propose a sampling method to calculate the blocking value $B(C,d)$ based on the LE sub-graph under the extended LT model.

3.2. Competitive influence propagation in linear threshold model

We investigate the USN-IBM problem under the linear threshold (LT) model. In the model, each node v is assigned an activation threshold $\Theta_v \in [0, 1]$. Let $N(v)$ be the set of node v's in-neighbors in the network. A real number $P(u,v)$ is assigned on each edge (u,v) indicating the probability for the influence being propagated through the edge. The probability on the edges connected with each node v satisfies

$$\sum_{u \in N(v)} P(u, v) = 1$$

Under the LT model, each node is either influenced or uninfluenced by the seeds at each time step. Initially, all the seed nodes are treated as the influenced ones and the others are uninfluenced. We use $Na(v)$ to denote the set of influenced in-neighbors of an uninfluenced node v. If v satisfies $\sum_{u \in Na(v)} P(u, v) \ge \theta_v$, then it will be influenced. After node v being influenced, it attempts to influence its uninfluenced out-neighbors. Such process of influence propagation will end when there is no node to be influenced any more.

We use the network in Fig. 1 as an example. The independent activation thresholds of nodes a, b, c and d are denoted as $\theta_a, \theta_b, \theta_c, \theta_d$ respectively, and their values are all set as 0.5. On the edges, the influence propagation probabilities are $P(a,b)$=0.6, $P(a,c)$=$P(b,c)$=$P(a,d)$=0.3. Let the seed set be $\{a\}$. At time T, seed a has been influenced. It attempts to influence its uninfluenced out-neighbors b, c and d at time T+1. Since $P(a,b)$=0.6> θ_b, node a successfully influences its neighbor b. But nodes c and d are not influenced, since $P(a,c)$=$P(a,d)$=0.3 are less than θ_c and θ_d. At time $t = T + 2$, since both a and b are influenced, they attempt to influence their out-neighbor node c. The probability of influencing node c is $P(a,c)$+$P(b,c)$=0.6, which is larger than θ_c. Therefore, node c is influenced. Such procedure of influence propagation ends at time T+3, since no more nodes can be influenced. Eventually, nodes a, b, c are influenced, and d is uninfluenced.

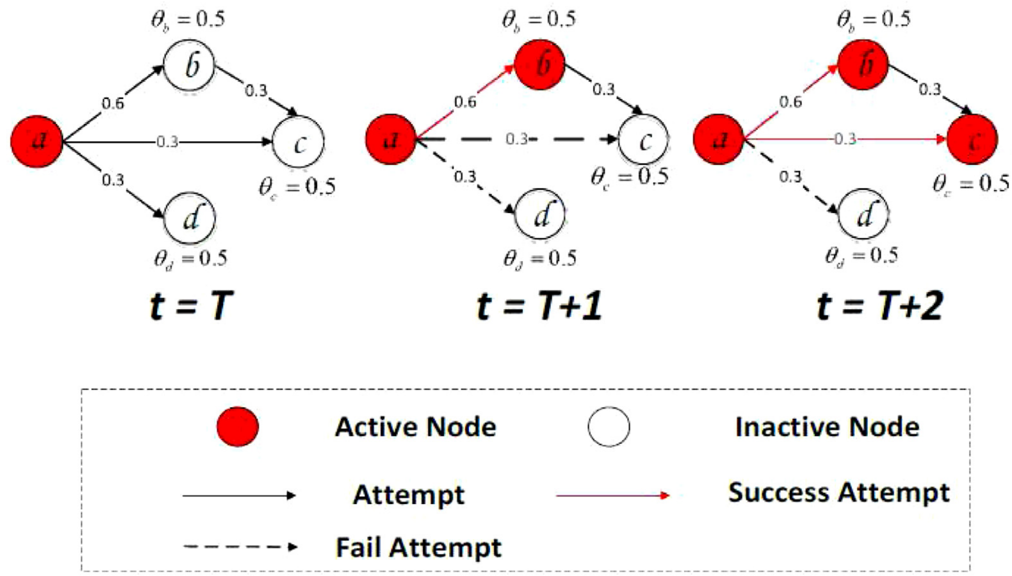

Fig. 1. The Linear Threshold Model.

One advantage of the linear threshold model (LT) model is that it postulates that different nodes in the network have different thresholds to be activated. Other models, such as the IC model, do not have such threshold for activating each node, they implicitly assume that all the nodes have identical activation threshold. Therefore, LT model can more accurately reflect the nodes' states under the influence propagated in the social network. The other advantage of the LT model is that whether a node is activated depends not on the influence of one neighbor, but on the combined influence of all its neighbors. In the other models, such like the IC model, a node is activated by only one of its neighbors, not by the joint effect of all its neighbors.

For solving the UNS-IBM problem, we choose the LT propagation model for two reasons. First, the LT model is suitable for describing the propagation of two opposite influences. In this model, activating a node depends on the combined influence from all its neighbors, the combined effects of both positive and negative influences are taken into consideration. The second, under the LT mode, the influence propagation can be simulated by "live-edge" (LE) sub-graph, we can perform the node deletion on the propagation tree in the LE sub-graph to estimate the blocking increments.

The traditional LT propagation model is only for the cases where there is only positive influence in the network. To solve the problem of negative influence blocking, we extend the LT propagation model for the competitive networks which consist of both positive and negative influences. We define the competitive influence linear threshold propagation model (CI-LTPM) as follows.

Definition 4 (*Competitive Influence Linear Threshold Propagation Model (CI-LTPM)*)**.** Let $G = (V, E, P)$ beacompetitive network where both positive and negative influences are propagated simultaneously. Each edge (u, v) in G has a propagation probability $P(u, v)$. Each node v is assigned an activation threshold θ_v in the interval $[0, 1]$. Let C and D be the negative and positive seed sets respectively. At each propagation step of the CI-LTPM model, a node may stay in one of the following three states: If a node has not been influenced, it is in the uninfluenced state. Once it is influenced, it is in the positively or negatively influenced state. Node's state is changed by the propagation of positive and negative influences according to the following rules:

(1) Initially, all the positive and negative seed nodes are in the positively and negatively influenced states respectively, and all the other nodes are in the uninfluenced state.
(2) Let v be an uninfluenced node and $P\text{-}Act(v)$ be the set of v's in-neighbors which are in the positively influenced state. If node v satisfies: $\sum_{u\in P-Act(v)} P(u, v) \geq \theta_v$, v will change its state from uninfluenced to positively influenced.
(3) Let v be an uninfluenced node and $N\text{-}Act(v)$ be the set of v's in-neighbors which are in the negatively influenced state. If node v satisfies: $\sum_{u\in N-Act(v)} P(u, v) \geq \theta_v$, v will change its state from uninfluenced to negatively influenced.
(4) If node v can be influenced by both positive and negative influences at the same time, v will be negatively influenced according to the principle of "negative influence dominating" in social psychology.
(5) After a node being positively or negatively influenced, it cannot be influenced by the opposite influence any more.
(6) The procedure of propagation ends when there is no more nodes to be influenced.

3.3. Constructing the live-edge graph

Under the LT model, the live-edge (LE) sub-graph can be used to simulate the influence propagation. To construct the LE sub-graph, one incoming edge (u,v) of each node v is chosen to join the LE sub-graph according to the propagation probability $P(u,v)$. In the network $G = (V, E)$, suppose the set of live-edges selected is $E' \subseteq E$, then the LE sub-graph is $G' = (V, E')$.

We simulate the propagation of negative influence by generating multiple LE sub-graphs under the CI-LTPM model. By deleting some nodes from each LE sub-graph, we can calculate the negative influence blocking effect of the deleted nodes. Thus, the node set with the largest blocking effect will be selected as the positive seed set.

To solve the problem of UNS-IBM, the key issue is how to estimate the value of $B(C,d)$ in Eq. (5), which is the blocking effects of the positive seed set C on the negative influence by negative seed d. Based on Eq. (1), we have:

$$B(C, d) = \delta(V, d) - \delta(V\backslash C, d) \tag{6}$$

Here, $\delta(V\backslash C, d)$ is the negative influence spreading by the negative seed d under the blocking effect by the positive seed

```
Algorithm 1 Construct_LE(G)
Input: G = (V, E, P): direct graph;
Output: G': LE sub-graph;
        P(G'|G): probability for the sub-graph G' being constructed from G;
Begin
  E_k = ∅;   P(G'|G)=1;
  For each node v in V do
      Select an in-edge (u, v) of v according to P(u, v);
      If no edge is selected then
          P(G'|G)= P(G'|G)*[1 − Σ_{u∈N(v)} P(u, v)] ;
      else      /* let the selected edge be (u,v)*/
          E_k = E_k ∪ {(u, v)};
          P(G'|G)= P(G'|G)*P(u, v);
      End if;
  End for v;
  G' = (V, E_k);
  Return (G', P(G'|G));
End
```

Fig. 2. Algorithm Construct_LE.

set C. Under the CI-LTPM model, if there is no positive seed set, $\delta\,(V,d)$ is equal to the expectation of the number of the nodes reachable by d in all the LE sub-graphs.

Let $W(G)$ be the set of all the LE sub-graphs of G, $G' \in W\,(G)$ be a LE sub-graph of G, and $\delta_{G'}(V,d)$ be the number of nodes reachable by d in G', then we have:

$$\delta(V,d) = \sum_{G'\in W(G)} \Pr\left(G'|G\right) * \delta_{G'}(V,d) \tag{7}$$

where $Pr\left(G'|G\right)$ indicates the probability for G' being constructed in G, which can be calculated by the following equation:

$$\Pr\left(G'|G\right) = \prod_{d\in V} p(d, G', G) \tag{8}$$

Here, $p\left(d, G', G\right)$ is the probability for node d connecting with at least one edge in G'. Let $N(d)$ be the set of d's in-neighbors, $P(u,d)$ be the propagation probability on edge (u,d). Then $p(d, G', G)$ can be calculated by

$$p\left(d, G', G\right) = \begin{cases} P\,(u,d) & \text{edge } (u,d) \text{ is selected in } G' \\ 1 - \sum_{u\in N(d)} P\,(u,d) & \text{otherwise} \end{cases} \tag{9}$$

We propose an algorithm named *Construct_LE* to construct the LE sub-graph G' from G. For each node v of G, the algorithm selects one in-edge (u,v) connecting with v according to the probability $P(u,v)$. According to (8), Algorithm *Construct_LE* also computes $P(G'|\,G)$ which is the probability for the LE sub-graph G' being constructed from G. Framework of the algorithm *Construct_LE* is shown in Fig. 2.

Let m be the number of edges in $G = (V,E)$. For each node v in G, Algorithm *Construct_LE* selects one of its in-edge to join the LE sub-graph time, and calculates the probability of the LE sub-graph using the probabilities of the edges linking with v. Therefore, time complexity for Algorithm *Construct_LE* is $O(m)$.

4. Calculating the blocking values based on propagation tree

To solve the problem of *UNS-IBM*, we must estimate the value of $B(C,d)$ in Eq. (5), which is the blocking effect by positive seed set C on the negative influence from seed d. From (6), we know that the value of $B(C,d)$ can be estimated by $\delta\,(V,d)$-$\delta\,(V\backslash C,d)$ which is the reduction of the negative influence spreading by positive seed set C. We notice that if we remove a node c and its related links from the network, the negative influence cannot be propagated via node c anymore. Let $b(c)$ be the reduction of negative influence after deleting node c. Suppose c is a positive seed, then the blocking effect by c is $b(c)$. Therefore, the negative influence blocking by each positive seed node c is equal to the reduction of the negative influence after removing c and its related links. Based on such observation, we propose a node deletion-based algorithm for blocking the negative influence from uncertain sources. We simulate the propagation of negative influence by generating multiple LE sub-graphs. In each LE sub-graph, we define the propagation tree rooted at each negative seed d, and estimate the increment of the influence blocking value of each candidate positive seed c by removing it from the propagation tree. Finally, the positive seed set can be selected according to the increment of the blocking effect of each node.

4.1. Propagation tree and its coverage

To calculate the blocking effect $B(C,d)$ by (6), we must estimate the values of $\delta\,(V,d)$ and $\delta\,(V\backslash C,d)$. By (7) we know that $\delta\,(V,d)$ can be calculated based on $\delta_{G'}\,(V,d)$, which can be computed in the LE sub-graph G' by the following equation:

$$\delta_{G'}\,(V,d) = \sum_{u\in V} I(u,d,G') \tag{10}$$

where

$$I\left(u,d,G'\right) = \begin{cases} 1 & \text{exist a route from } u \text{ to } d \text{ in } G' \\ 0 & \text{otherwise} \end{cases}$$

We denote the set of all the LE sub-graphs in graph G as $W(G)$. By combining (7) with (10), we can get:

$$\begin{aligned} \delta\,(V,d) &= \sum_{G'\in W(G)} \Pr\left(G'\right) * \delta_{G'}(V,d) \\ &= \sum_{G'\in W(G)} \sum_{u\in V} I(u,d,G') * \Pr\left(G'\right) \\ &= \sum_{G'\in W(G)} \Pr\left(G'\right) \sum_{u\in V} I(u,d,G') \end{aligned} \tag{11}$$

In order to estimate $I\left(u,d,G'\right)$ in (11) for each possible negative seed d, we define the propagation tree rooted at d in the LE sub-graph G'.

Definition 5 (*Propagation Tree*)**.** The propagation tree of a node d in LE sub-graph G', which is denoted as $T_{G'}^{d}$, is defined as the breadth-first-search tree rooted at d in G'.

Since each node in the LE sub-graph is connected with only one in-coming edge, such propagation tree rooted at d uniquely exists in G'. In Section 4.3, we will present an algorithm for constructing the propagation tree.

Based on the propagation tree rooted at a negative seed d, we can estimate the propagation function $\delta\,(V,d)$ as

$$\delta\,(V,d) = \sum_{G'\in W(G)} \Pr(G') \sum_{u\in V} I(u, T_{G'}^{d}) \tag{12}$$

Here,

$$I(u, T^d_{G'}) = \begin{cases} 1 & \text{exists a route from } u \text{ to } d \text{ in } T^d_{G'} \\ 0 & \text{otherwise} \end{cases} \quad (13)$$

Definition 6 (*Coverage of Node u in $T^d_{G'}$*)**.** The coverage of node u in $T^d_{G'}$, which is denoted as $r\left(u, T^d_{G'}\right)$, is the number of nodes in the sub-tree rooted at u in $T^d_{G'}$:

$$r\left(u, T^d_{G'}\right) = \sum_{x \in T^d_{G'}} I\left(x, T^u_{G'}\right) \quad (14)$$

If we substitute u in (14) with d, we can see that $r\left(d, T^d_{G'}\right)$ is just the number of nodes in $T^d_{G'}$ namely

$$r\left(d, T^d_{G'}\right) = \sum_{u \in V} I(u, T^d_{G'}) \quad (15)$$

(15)

Combining (12) and (15), we can get

$$\delta\left(V, d\right) = \sum_{G' \in W(G)} r(d, T^d_{G'}) * \Pr(G') \quad (16)$$

4.2. Estimating the increment of blocking value

Suppose that a candidate positive seed node w is added into the partial positive seed set C. To calculate the increment of blocking value by the new seed w, we delete it and its related edges from G to get a sub-graph which is denoted as $G\left(V \backslash \{w\}\right)$. Then we can estimate the negative influence spreading in $G\left(V \backslash \{w\}\right)$. For each LE sub-graph G' of G, there exists an unique LE sub-graph $G'_{G(V\backslash\{w\})}$ of $G\left(V \backslash \{w\}\right)$ corresponding to G'. Let d be a negative seed, and $T^d_{G'}$ be the tree rooted at v in G', $G'_{G(V\backslash\{w\})}$ also has a propagation tree $T^d_{G'(V\backslash\{w\})}$ rooted at v corresponding to $T^d_{G'}$ in G'. Then the negative influence spreading by d in LE sub-graph $G'_{G(V\backslash\{w\})}$ is $r\left(d, T^d_{G'_{G(V\backslash\{w\})}}\right)$. The difference between $r\left(d, T^d_{G'}\right)$ and $r\left(d, T^d_{G'_{G(V\backslash\{w\})}}\right)$ is the blocking effect by w on the negative influence propagated from d in G'. Comparing with the propagating tree $T^d_{G'}$, we can see that only node w and the sub-tree rooted at w are missing in $T^d_{G'_{G(V\backslash\{w\})}}$. Therefore, it is obvious that:

$$r\left(d, T^d_{G'}\right) - r\left(d, T^d_{G'_{G(V\backslash\{w\})}}\right) = r\left(w, T^d_{G'}\right) + 1 \quad (17)$$

Suppose C is a partial positive set, and d is a negative seed, after adding a candidate positive seed w to C, the blocking increment on the negative influence of d is:

$$r\left(d, T^d_{G'_{G(V\backslash C)}}\right) - r\left(d, T^d_{G'_{G(V\backslash(C\cup\{w\}))}}\right) = r\left(w, T^d_{G'_{G(V\backslash C)}}\right) + 1 \quad (18)$$

Given positive and negative seed sets C and D, and : a candidate positive seed $w \in V \backslash D \backslash C$, after adding node w to C, the blocking increment value $\Delta\left(w\right)$ is:

$$\begin{aligned} \Delta\left(w\right) &= B\left(C \cup \{w\}, D\right) - B\left(C, D\right) \\ &= \left[\delta\left(V, D\right) - \delta\left(V(C \cup \{w\}), D\right)\right] - \left[\delta\left(V, D\right) - \delta\left(V \backslash C, D\right)\right] \\ &= \delta\left(V \backslash C, D\right) - \delta\left(V(C \cup \{w\}), D\right) \\ &= \sum_{d \in D} \delta(V \backslash C, d) - \sum_{d \in D} \delta\left(V \backslash (C \cup \{w\}), d\right) \\ &= \sum_{d \in D} \sum_{G' \in W(V\backslash C)} \left[r\left(d, T^d_{G'_{G(V\backslash C)}}\right) - r\left(d, T^d_{G'_{G(V\backslash(C\cup\{w\}))}}\right)\right] * \Pr(G') \end{aligned}$$

Therefore, by (18) we have

$$\Delta\left(w\right) = \sum_{d \in D} \sum_{G' \in W(V\backslash C)} \left[r\left(w, T^d_{G'_{G(V\backslash C)}}\right) + 1\right] * \Pr(G') \quad (19)$$

Similarly, in the case of uncertain negative seed set, for a positive seed node set C, after adding a candidate positive seed w to C, the blocking increment $\Delta(w)$ can be estimated by

$$\begin{aligned} \Delta\left(w\right) &= \sum_{d \in V} P^{-}(d) * \sum_{G' \in W(V\backslash C)} \left[r\left(w, T^d_{G'_{G(V\backslash C)}}\right) + 1\right] * \Pr(G') \\ &= \sum_{G' \in W(V\backslash C)} \sum_{d \in V} \left[r\left(w, T^d_{G'_{G(V\backslash C)}}\right) + 1\right] P^{-}\left(d\right) * \Pr(G') \end{aligned} \quad (20)$$

From (20), we can see that the blocking increment $\Delta\left(w\right)$ of the candidate positive seed w can be calculated by constructing LE sub-graphs $G'_{G(V\backslash C)}$ from G. The blocking increment $\Delta\left(w\right)$ is the weighted summation of w's coverages in all the propagation trees $T^d_{G'_{G(V\backslash C)}}$ for $d \in V \backslash C$. The node with the highest blocking increment $\Delta(w)$ will be chosen as the new positive seed.

4.3. Constructing the propagation tree

To calculate the blocking increment $\Delta(w)$ for each node w, we present an algorithm named *Construct_PT* (construct propagation tree) to construct the propagation tree $T^v_{G'}$ of each node v in G'. Based on the propagation tree $T^v_{G'}$ the value of $r\left(v, T^v_{G'}\right)$ and the blocking increment $\Delta\left(w\right)$ of w in $T^v_{G'}$ can be calculated. The algorithm takes each node v as the root and uses the breadth-first-search method to generate the propagation tree $T^v_{G'}$. The propagation tree $T^v_{G'}$ is represented by recording the parent and children of each node in the tree. After constructing the propagation tree $T^v_{G'}$, we calculate the coverage $r\left(u, T^v_{G'}\right)$ of each node u in $T^v_{G'}$, which is the number of the offsprings of u in $T^v_{G'}$. The partial value of $\Delta(u)$ in G' is $\sum_{v \in D}\left[r\left(u, T^v_{G'_{G(V\backslash C)}}\right) + 1\right] P^{-}\left(v\right) * \Pr(G')$. Framework of the algorithm *Construct_PT* is shown in Fig. 3.

Let n and m be the numbers of nodes and edges in $G = (V,E)$ respectively. Algorithm *Construct_PT* constructs propagation tree $T^v_{G'}$ for all the nodes in G'. A breadth-first-search method is used to construct the propagation tree $T^v_{G'}$ in O(m) time. Therefore, time complexity of Algorithm *Construct_PT* is O($m.n$).

4.4. Estimating the number of LE sub-graphs

It can be seen from (20) that we need to generate all the LE sub-graphs in $W\left(G\right)$ to calculate the blocking increment $\Delta\left(w\right)$ for each candidate positive seed w. However, to build the propagating trees in all the LE sub-graph requires huge amount of time. To overcome this obstacle, we present a sampling approach which randomly chooses part of the LE sub-graphs to construct a subset of $W(G)$. Let $H(G)$ be the set of the LE sub-graphs randomly chosen from $W(G)$. We can approximate the value of $\Delta\left(w\right)$by replacing $W(G)$ in (20) with $H(G)$. Obviously, the more LE sub-graphs we build, the more accurate blocking increment we can get, but the more computation time is required. To balance the computation time and the accuracy of the results, we predefine an error threshold ε, and a probability ρ. To restrict the error within ε under a probability ρ, we use Chernoff bound to estimate the number of the sampled LE sub-graphs in $H\left(G\right)$.

Theorem 1 (*Chernoff Bound*)**.** *Given R independent and identically distributed random variables $X_1, X_2, \ldots, X_R$ in [0,1] with mean μ.*

Algorithm 2 Construct_PT (G_i)

Input: G_i: a LE sub-graph of G;
L: max length of propagation path;
Output: $Parent(v,w,G_i)$: the parent node of node w in the sub-tree $T_{G_i}^v$;
$Children(v,w,G_i)$: the child nodes set of node w in the sub-tree $T_{G_i}^v$;
$\Delta(v)$: the blocking increment of node v;
Begin
Initialize the value of all $r(w,T_{G_i}^v)$ as 0 ;
For each node v in V **do**
$u=v$; $l=0$; $Q=\emptyset$; $H=\emptyset$; /*Q is a queue; H is a stack.*/
Repeat
For each of u's unvisited neighbor w **do**
Put $(w,l+1)$ to the tail of queue Q;
Put $(w,l+1)$ to the top of stack H;
$Parent(v,w,G_i)=u$;
$Children(v,u,G_i)=Children(v,u,G_i)\cup\{w\}$;
End for w;
(u,l) ←the head of queue Q;
Until $Q=\emptyset$ **or** $l=L$;
While $H\neq\emptyset$ **do**
(u,l) ←the top of stack H;
$r(parent(v,u,G_i),T_{G_i}^v)=r(parent(v,u,G_i),T_{G_i}^v)+[r(u,T_{G_i}^v)+1]*\Pr(G_i)*P^-(u)$;
$\Delta(parent(v,u,G_i))=\Delta(parent(v,u,G_i))+[r(u,T_{G_i}^v)+1]*\Pr(G_i)*P^-(u)$;
End While;
End for v;
End

Fig. 3. Algorithm Construct_PT.

Let X be the summation of $X_1, X_2, \ldots, X_R$. For a given error threshold $\epsilon > 0$, we have

$$\Pr(|X-R\mu|\geq\varepsilon R\mu)\leq e^{\frac{\varepsilon^2 R\mu}{2+\varepsilon}}.$$

By Theorem 1, we can calculate the number of sampled LE sub-graphs for a given error threshold ε. Suppose the LE sub-graphs sampled from G are $G_1, G_2, \ldots, G_R$. Let $f(G_i)=\max_{u,w\in V}\left[r\left(w,T_{G_i}^u\right)+1\right]$ be the largest negative influence spreading in G_i, we can get R random variables $X_i=\frac{1}{n}f(G_i)(i=1,2,\ldots,R)$. Since the values of $f(G_i)$ are in [0, n], the values of X_i are in [0,1]. Let $X=\sum_{i=1}^{R}f(G_i)/n$, its expectation is $\mu=E[X]=r^*/n$, where $r^*=\max_{G_i\in W(G)}\left[r\left(w,T_{G_i}^u\right)+1\right]$. Given an error threshold ε, by Theorem 1 we get

$$\Pr(|X-R\mu|\geq\varepsilon R\mu)=\Pr\left[\left|\frac{1}{n}\sum_{i=1}^{R}f(G_i)-\frac{R.r*}{n}\right|\geq\varepsilon\frac{R.r*}{n}\right]$$

$$=\Pr\left[\left|\sum_{i=1}^{R}f(G_i)-R.\,r*\right|\geq\varepsilon R.r*\right]\leq e^{\frac{\varepsilon^2 Rf^*}{(2+\varepsilon)n}} \tag{21}$$

By (21), we know that when the number of LE sub-graphs sampled from G satisfies:

$$R\geq\frac{(\varepsilon+2)n}{\varepsilon^2 r^*}\ln\frac{1}{\rho}, \tag{22}$$

Algorithm 3 *NDB* (G)

Input: $G=(V,E,P)$: Direct graph;
P^-: Distribution of negative seeds;
k: Size of positive seeds set;
Output: C: Set of positive seeds;
Begin
Use (23) to calculate the size of sampling R;
For i= 1 **to** R **do**
/*Construct R LE sub-graph */
$G_i=Construct\_LE(G)$;
/*Construct propagating trees in G_i and initialize $\Delta(u)$ for all the nodes*/
$T_i=Construct\_PT(G_i)$;
End for i;
$C^*=\emptyset$;

For t=1 **to** k **do**
$v=\operatorname{argmin}_{u\in V\setminus C}\Delta(u)$;
$C=C\cup\{v\}$;
For i= 1 **to** R **do** /*for the R LE sub-graphs*/
For each node u in G_i **do**
$w=parent(u,v,G_i)$;
While $w\neq u$ **do**
$r(w,T_{G_i}^u)=r(w,T_{G_i}^u)-[(r(v,T_{G_i}^u))+1]*\Pr(G_i)*P^-(u)$;
$\Delta(parent(u,w,G_i))=\Delta(parent(u,w,G_i))-[r(v,T_{G_i}^u)+1]*\Pr(G_i)*P^-(u)$;
$w=parent(u,w,G_i)$;
End while;
Put v into the tail of queue Q; $visited=\{v\}$;
While queue $Q\neq\emptyset$ **do**
w← head of queue Q;
For each $x\in Children(u,w,G_i)\setminus visited$ **do**
$visited=visited\cup\{x\}$;
Put x to the tail of queue Q;
$\Delta(x)=\Delta(x)-[r(x,T_{G_i}^u)+1]*\Pr(G_i)*P^-(u)$;
$r(x,T_{G_i}^u)=0$;
End for x;
End while
End for u
End for i
End for k
End

Fig. 4. Algorithm Node-Deletion Blocking.

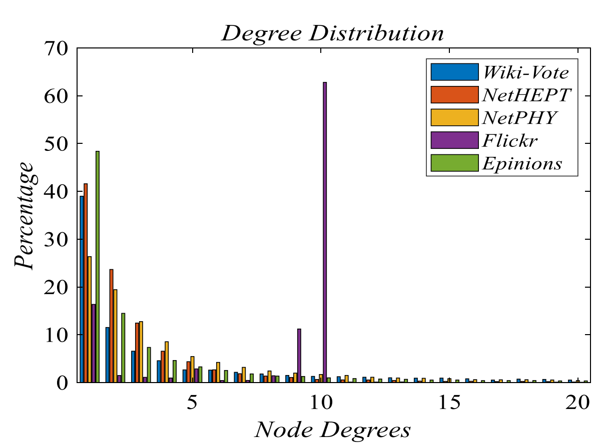

Fig. 5. Degree distribution of the networks.

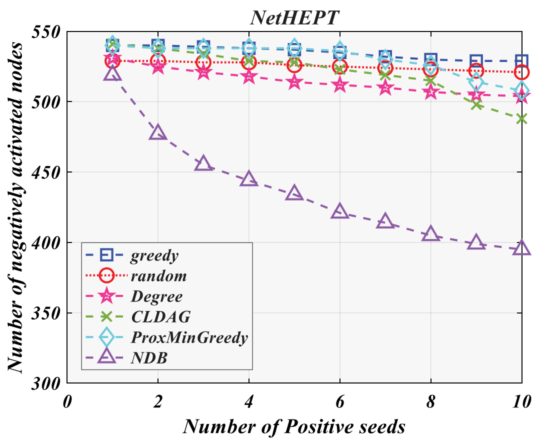

Fig. 6. Negative Influence Blocking by the algorithms on NetHEPT.

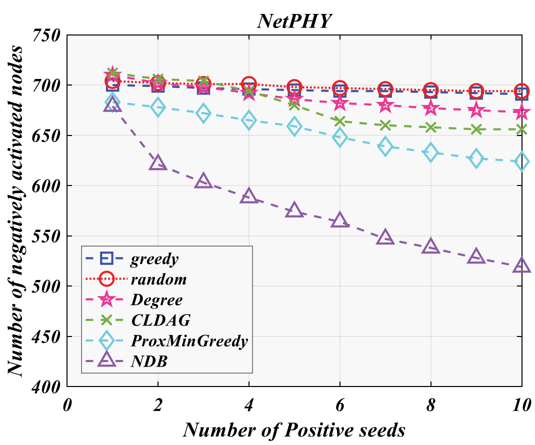

Fig. 7. Negative Influence Blocking by the algorithms on NetPHY.

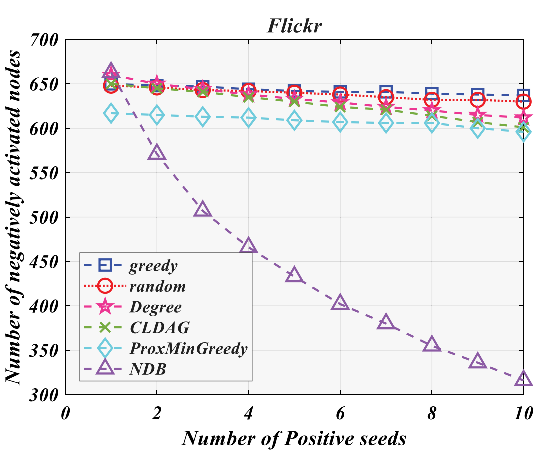

Fig. 8. Negative Influence Blocking by the algorithms on Flickr.

the error will be less than ε under the probability ρ.

To calculate the number of samplings by (22), we must know the value of r^*, which is the maximal negative influence spreading in each LE sub-graph. Suppose the maximum size of a negative seed set is N_D. Since the influence spreading $r^* \geq N_D$, it is easy to know

$$\frac{(\varepsilon+2)n}{\varepsilon^2 N_D}\ln\frac{1}{\rho} \geq \frac{(\varepsilon+2)n}{\varepsilon^2 r^*}\ln\frac{1}{\rho}$$

Using N_D to substitute r^* in (22), we get the following theorem.

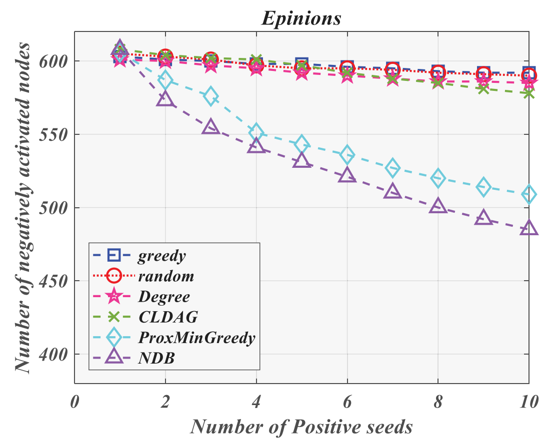

Fig. 9. Negative Influence Blocking by the algorithms on Epinions.

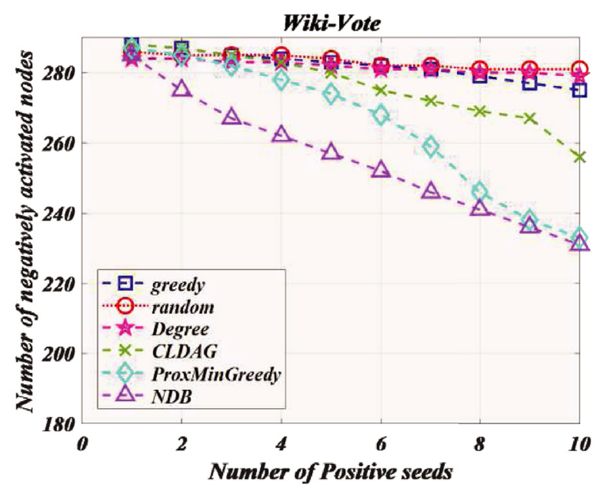

Fig. 10. Negative Influence Blocking by the algorithms on Wiki-Vote.

Theorem 2. *Let the maximum size of a negative seed set be N_D. Given a probability ρ and an error threshold ε, if the number of sampled LE sub-graphs satisfies:*

$$R \geq \frac{(\varepsilon+2)\,n}{\varepsilon^2 N_D}\ln\frac{1}{\rho} \tag{23}$$

the error will be less than ε under the probability ρ.

5. Maximizing the blocking on negative influence from uncertain sources

In this section, we first analysis the properties of LE sub-graphs, and then prove that the negative influence blocking function $B(C)$ based on the LE sub-graphs under the CI-LTPM model is monotonic and sub-modular with respect to C. Since the objective function $B(C)$ is sub-modular, we can use the greedy approach for positive seed detection. As a result, a node deletion-based algorithm is presented for the UNS-IBM problem.

5.1. Properties of the LE sub-graph and blocking function

Since the negative influence blocking function $B(C)$ in (5) is a linear combination of $B(C,d)$ over all the possible negative seeds, we need only to prove that $B(C,d)$ is monotonic increasing and sub-modular with respect to C. Noticing that $B(C,d) = \delta(V,d) -$

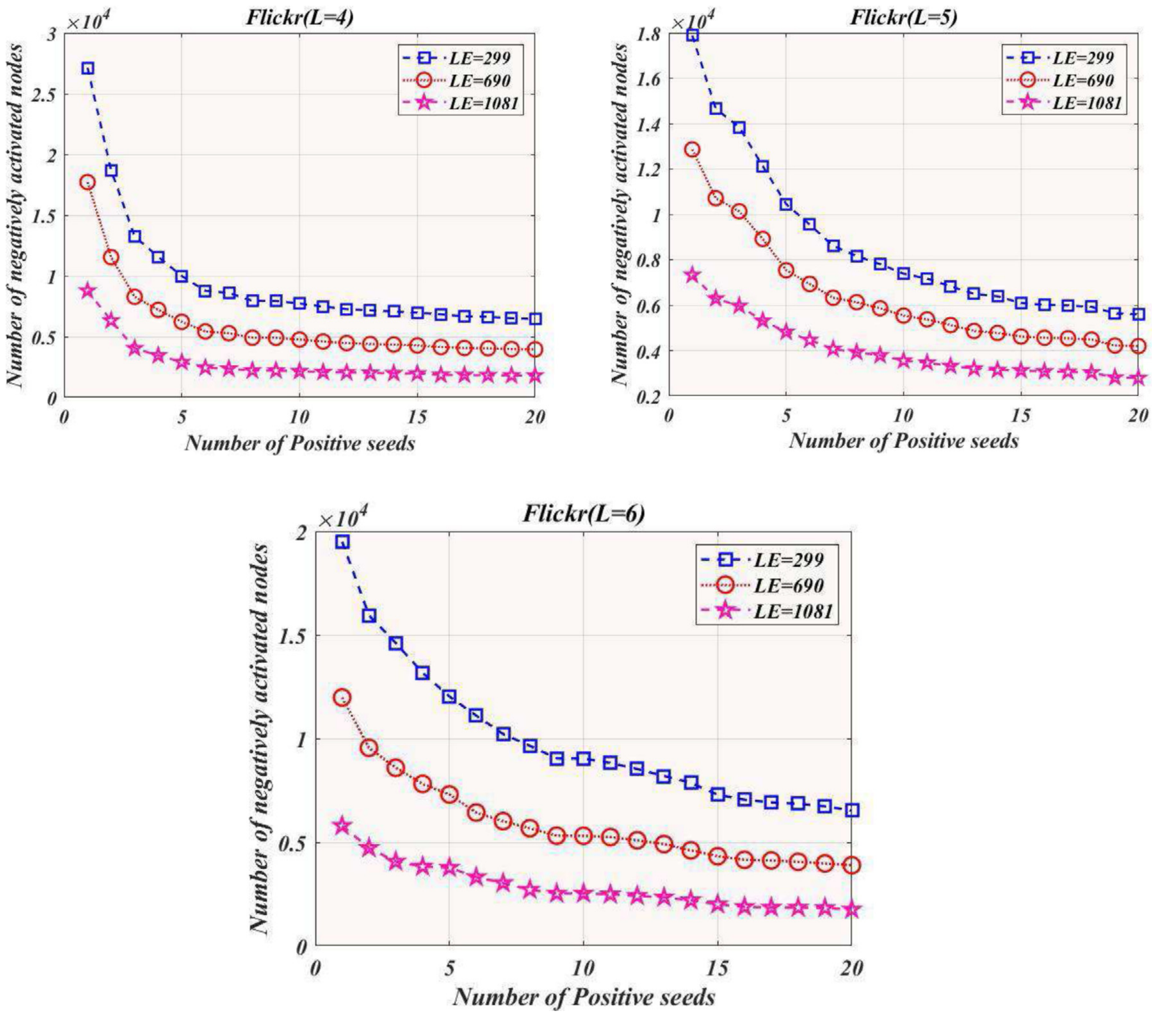

Fig. 11. Negative Influence Blocking on Flickr by different numbers of LE sub-graphs.

$\delta\left(V\backslash C, d\right)$, we need to prove that $\delta\left(V\backslash C, d\right)$ is monotonically decreasing and super-modular with respect to C.

Let E' be the set of edges in G which are connected with positive seeds in C. In the following sections of this paper, we use $G\backslash C$=$\left(V\backslash C, E\backslash E'\right)$ to denote the sub-graph obtained by deleting the nodes in C and their related edges from G.

To prove the monotonic and super-modular properties of $\delta(V\backslash C, u)$, we first give the following lemmas to show some properties of the LE sub-graph.

Lemma 1. *Suppose that the in-edge linking with node u in LE sub-graph G' is (v, u), then*

$$\Pr\left(G'|G\backslash C\right) = \Pr(G'|G\backslash(C\cup\{u\})) * P(v, u)$$

Proof. By Eq. (8), we have:

$$\Pr\left(G'|G\backslash C\right) = \prod_{w\in V\backslash C} p(w, G', G)$$
$$= p\left(u, G', G\right) * \prod_{w\in V\backslash(C\cup\{u\})} p(w, G', G).$$

By Eq. (9), we know

$$\Pr\left(G'|G\backslash C\right) = P(v, u) * \prod_{w\in V\backslash(C\cup\{u\})} p(w, G', G)$$
$$= \Pr(G'|G\backslash(C\cup\{u\})) * P(v, u). \quad \square$$

Lemma 2. *If node u is linked with an in-edge (v, u) in LE sub-graph G', then*

$$\Pr\left(G'|G\backslash C\right) - \Pr(G'|G\backslash(C\cup\{u\}))$$
$$= [p\left(v, u\right) - 1] * \sum_{v'\in\in V\backslash(C\cup\{u\})} p(v', G', G\backslash C)$$

Proof. By (8), we know:

$$\Pr\left(G'|G\backslash C\right) - \Pr(G'|G\backslash(C\cup\{u\}))$$
$$= \prod_{v\in V\backslash C} p\left(v, G', G\backslash C\right) - \prod_{v'\in V\backslash(C\cup\{u\})} p(v', G', G\backslash(C\cup\{u\}))$$

For each node $v' \in V\backslash\{u\}$, we have $p(v', G', G\backslash(C\cup\{u\})) = p(v', G', G\backslash C)$. Then

$$\Pr\left(G'|G\backslash C\right) - \Pr(G'|G\backslash(C\cup\{u\}))$$
$$= p\left(v, u\right) * \prod_{v'\in V\backslash(C\cup\{u\})} p\left(v', G', G\backslash C\right) - \prod_{v'\in\in V\backslash(C\cup\{u\})} p(v', G', G\backslash C)$$
$$= [p\left(v, u\right) - 1] * \prod_{v'\in\in V\backslash(C\cup\{u\})} p(v', G', G\backslash C) \quad \square$$

Let $W\left(G\backslash C\right)$ be the set of all LE sub-graphs of $G\backslash C$, and v be a node in $V\backslash C$. We divide the LE sub-graphs in $W\left(G\backslash C\right)$ into two groups as follows:

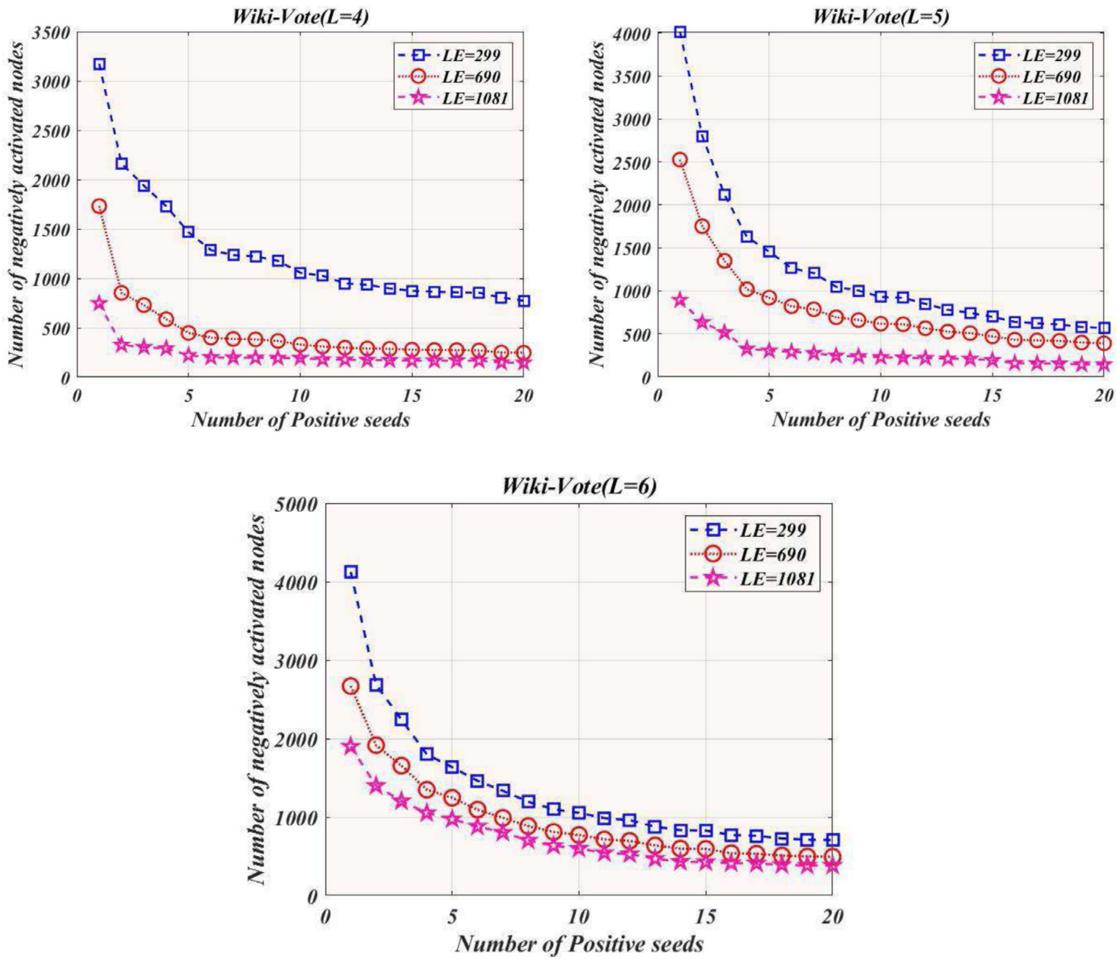

Fig. 12. Negative Influence Blocking onWiki-Vote by different numbers of LE sub-graphs.

The first group is denoted as $W(G\backslash C, v)$, which consists of the LE sub-graphs with node v and its related edges.

The second group is denoted as $W(G\backslash C, \overline{v})$, which consists of the LE sub-graphs without node v and its related edges.

Lemma 3. *Suppose that node v in graph $G\backslash C$ is connected with m in-edges $e_1, e_2, \ldots, e_m$. For each LE sub-graph G' in $W(G\backslash C, \overline{v})$, there is a one-to-one correspondence between G' and a set of LE sub-graphs $Q(G') = \{G'_1, G'_2, \ldots, G'_m\}$ in $W(G\backslash C, v)$. Their probabilities satisfy*

$$\sum_{\mathrm{i}=1}^{m} \Pr\left(G'_i | G\backslash C\right) = \Pr\left(G' | G\backslash (C \cup \{\mathrm{v}\})\right)$$

Proof. Let G' be a LE sub-graph in $W(G\backslash C, \overline{v})$, where node v and its related in-edges are excluded. If we add node v and in-edge e_i to G', we can get a LE sub-graph G'_i in $W(G\backslash C, v)$. All the LE sub-graphs $G'_i\ (i = 1, 2, \ldots, m)$ form a set $Q(G') = G'_1, G'_2, \ldots, G'_m$ in $W(G\backslash C, v)$. Thus, G' is uniquely corresponding to the set $Q(G')$ in $W(G\backslash C, v)$.

On the other hand, let $Q(G') = \{G'_1, G'_2, \ldots, G'_m\}$ be a set of live-edge graphs in $W(G\backslash C, v)$, where each G'_i is constructed by adding node v and its related in-edge e_i in G'. Obviously set $Q(G')$ is uniquely corresponding to G' in $W(G\backslash C, \overline{v})$.

Therefore, there is a one-to-one correspondence between G' in$W(G\backslash C, \overline{v})$ and a set of LE sub-graphs $Q(G') = \{G'_1, G'_2, \ldots, G'_m\}$ in $W(G\backslash C, v)$.

For each G'_i in $W(G\backslash C, v)$, we have

$$P_r\left(G'_i | G\backslash C\right) = \prod_{w \in V\backslash C} p\left(w, G'_i, G\right) = p_r(e_i) \cdot \prod_{w \in V\backslash (C \cup \{v\})} p\left(w, G'_i, G\right)$$
$$= p_r(e_i) \cdot \prod_{w \in V\backslash (C \cup \{v\})} p\left(w, G', G\right).$$

Therefore

$$\sum_{\mathrm{i}=1}^{m} P_r\left(G'_i | G\backslash C\right) = \sum_{\mathrm{i}=1}^{m} p_r(e_i) \prod_{w \in V\backslash (C \cup \{v\})} p\left(w, G', G\right).$$

Since $\sum_{i=1}^{m} p_r(e_i) = 1$, we get

$$\sum_{\mathrm{i}=1}^{m} P_r\left(G'_i | G\backslash C\right) = \prod_{w \in V\backslash (C \cup \{v\})} p\left(w, G', G\right) = p_r\left(G'_i | G\backslash C\right) \quad \square$$

Theorem 3. *On the LE sub-graphs under the CI-LTPM model, the function $\delta(G\backslash C, u)$ is monotonically decreasing with respect to C.*

Proof. Let $\delta(u) = \delta(G\backslash C, u) - \delta(G\backslash (C \cup \{v\}), u)$. To prove that function $\delta(G\backslash C, u)$ is monotonically decreasing, we need to prove $\delta(u) \geq 0$. By (11), we have

$$\delta(u) = \sum_{G' \in W(G\backslash C)} \Pr\left(G' | G\backslash C\right) * r(u, G')$$

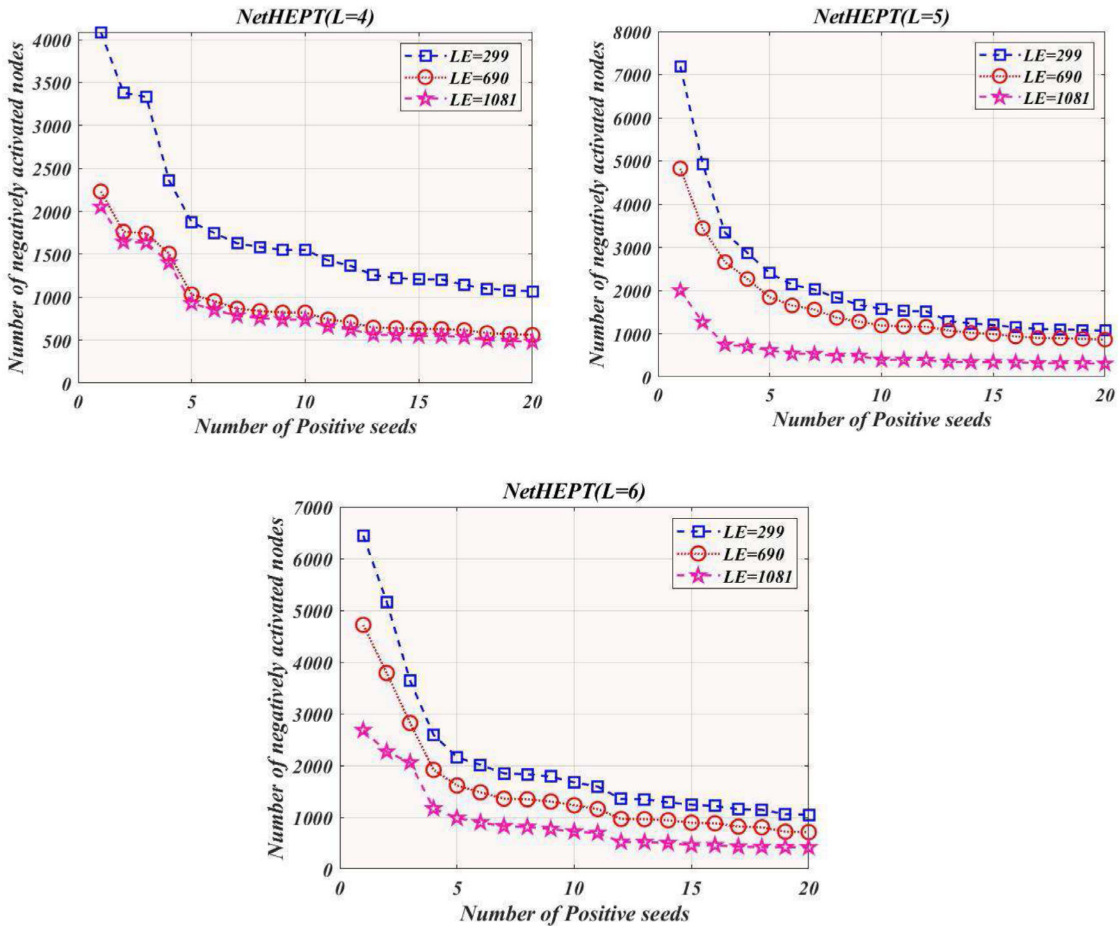

Fig. 13. Negative Influence Blocking on NetHEPTby different numbers of LE sub-graphs.

$$
\begin{aligned}
&- \sum_{G' \in W(G\backslash(C\cup\{v\}))} \Pr\left(G'|G\backslash(C\cup\{v\}) * r(u,G')\right) \\
&= \sum_{G'\in W(G\backslash C,v)} \Pr\left(G'|G\backslash C\right) * r(u,G') \\
&+ \sum_{G'\in W(G\backslash C,\overline{v})} \Pr\left(G'|G\backslash C\right) * r(u,G') \\
&- \sum_{G'\in W(G\backslash(C\cup\{v\}),v)} \Pr\left(G'|G\backslash(C\cup\{v\})\right) * r(u,G') \\
&- \sum_{G'\in W(G\backslash(C\cup\{v\}),\overline{v})} \Pr(G'|G\backslash(C\cup\{v\}) * r(u,G'))
\end{aligned}
$$

Since v does not exist in $G\backslash(C\cup\{v\})$, we have

$$
\sum_{G'\in W(G\backslash(C\cup\{v\}),v)} \Pr\left(G'|G\backslash(C\cup\{v\})\right) * r(u,G') = 0
$$

Then we get:

$$
\begin{aligned}
\delta(u) = &\sum_{G'\in W(G\backslash C,v)} \Pr\left(G'|G\backslash C\right) * r(u,G') \\
&+ \sum_{G'\in W(G\backslash C,\overline{v})} \Pr\left(G'|G\backslash C\right) * r(u,G') \\
&- \sum_{G'\in W(G\backslash(C\cup\{v\}),\overline{v})} \Pr(G'|G\backslash(C\cup\{v\})) * r(u,G')
\end{aligned}
$$

Because each LE sub-graph of $W(G\backslash(C\cup\{v\}),\overline{v})$ does not contain node v and its connected in-edges, we have $W(G\backslash C,\overline{v}) = W(G\backslash(C\cup\{v\}),\overline{v})$. Therefore:

$$
\begin{aligned}
\delta(u) = &\sum_{G'\in W(G\backslash C,v)} \Pr\left(G'|G\backslash C\right) * r(u,G') \\
&+ \sum_{G'\in W(G\backslash C,\overline{v})} \left[\Pr\left(G'|G\backslash C\right) - \Pr\left(G'|G\backslash(C\cup\{v\})\right)\right] * r(u,G')
\end{aligned}
$$

By Lemma 1, we have

$$
\begin{aligned}
\delta(u) = &\sum_{G'\in W(G\backslash C,v)} \Pr\left(G'|G\backslash C\right) * r(u,G') \\
&+ \sum_{G'\in W(G\backslash C,\overline{v})} \left[(p(u,v)-1) \prod_{v'\in V(C\cup\{v\})} p(v',G',G\backslash C)\right] \\
&* r(u,G') \\
= &\sum_{G'\in W(G\backslash C,v)} \Pr\left(G'|G\backslash C\right) * r(u,G') \\
&- \sum_{G'\in W(G\backslash C,\overline{v})} \Pr\left(G'|G\backslash C\right) * r\left(u,G'\right) \\
&+ p(u,v) \sum_{G'\in W(G\backslash C,\overline{v})} \left[\prod_{v'\in V(C\cup\{v\})} p\left(v',G',G\backslash C\right)\right] * r(u,G') \\
\geq &\sum_{G'\in W(G\backslash C,v)} \Pr\left(G'|G\backslash C\right) * r(u,G') \\
&- \sum_{G'\in W(G\backslash C,\overline{v})} \Pr\left(G'|G\backslash C\right) * r\left(u,G'\right)
\end{aligned}
$$

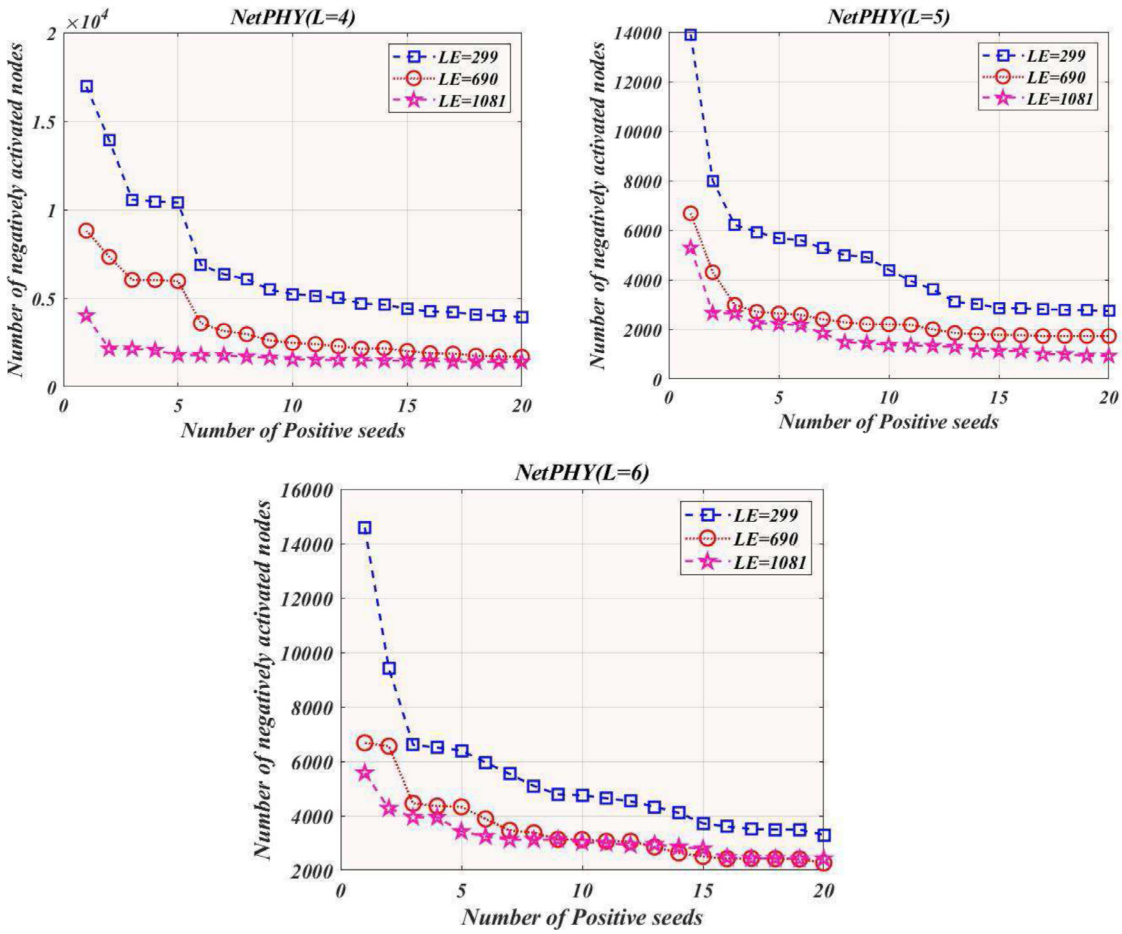

Fig. 14. Negative Influence Blocking on NetPHY by different numbers of LE sub-graphs.

By Lemma 3, we know that the LE sub-graph G' of $G\backslash(C\cup\{v\})$ uniquely corresponds to a group of LE sub-graphs $Q\left(G'\right) = \left\{G'_1, G'_2, \ldots, G'_m\right\}$ in $G\backslash C$. So we have

$$\delta\left(u\right) \geq \sum_{G'\in W(G\backslash C,\overline{v})} \sum_{i=1}^{m} \Pr\left(G'_i|G\backslash C\right) * r(u, G'_i) - \sum_{G'\in W(G\backslash C,\overline{v})} \Pr\left(G'|G\backslash C\right) * r\left(u, G'\right)$$

Let $r\left(u, G'_{min}\right) = \min_{1\leq i\leq m} r(u, G'_i)$, then

$$\begin{aligned}\delta\left(u\right) \geq & \sum_{G'\in W(G\backslash C,\overline{u})} \sum_{i=1}^{m} \Pr\left(G'_i|G\backslash C\right) * r\left(u, G'_{min}\right) \\ & - \sum_{G'\in W(G\backslash C,\overline{u})} \Pr\left(G'|G\backslash C\right) * r(u, G') \\ = & \sum_{G'\in W(G\backslash C,\overline{u})} \Pr(G'|G(C\cup u)) * r(u, G'_{min}) \\ & - \sum_{G'\in W(G\backslash C,\overline{u})} \Pr\left(G'|G\backslash C\right) * r(u, G')\end{aligned}$$

Because for each $G' \in W(G\backslash C, \overline{u})$, $\Pr\left(G'|G\backslash C\right) = \Pr(G'|G\backslash(C\cup u))$, we get

$$\delta\left(u\right) \geq \sum_{G'\in W(G\backslash C,\overline{u})} \sum_{i=1}^{m} Pr\left(G'_i|G\backslash(C\cup u)\right) * \left[r\left(u, G'_{min}\right) - r\left(u, G'\right)\right]$$

Since each G'_i has one more in-edge e_i than G', we have $r\left(u, G'_{min}\right) \geq r(u, G')$. Therefore

$$\delta\left(u\right) \geq \sum_{G'\in W(G\backslash C,\overline{u})} \sum_{i=1}^{m} \Pr\left(G'_i|G\backslash(C\cup u)\right) * \left[r\left(u, G'_{min}\right) - r\left(u, G'\right)\right] \geq 0$$

Thus, the function $\delta\left(G\backslash C, u\right)$ is monotonically decreasing with respect to C. □

Since $B\left(C, u\right) = \delta\left(V, u\right) - \delta\left(V\backslash C, u\right)$ and $\delta\left(V\backslash C, u\right)$ is monotonically decreasing with respect to C, and the negative influence blocking function $B(C)$ is a linear combination of $B\left(C, u\right)$, $B(C)$ in the LE sub-graphs is monotonically increasing with respect to C.

Definition 7 (*Super-modularity*)**.** Let $f(C)$ be a function of set, C and D be two sets such that $C \subseteq D \subset V$. Set function $f(C)$ is super-modular, if for each node $v \in V\backslash D\backslash C$, $f(C)$ satisfies

$$f\left(C\cup\{v\}\right) - f\left(C\right) \leq f\left(D\cup\{v\}\right) - f(D)$$

Theorem 4. *On the LE sub-graphs under the CI-LTPM model, the function $\delta(V\backslash C, u)$ is super-modular with respect to C.*

Proof. By (11), we know:

$$\delta\left(V\backslash C, u\right) = \sum_{G_i\in W(G\backslash C)} \delta_{G_i}\left(V\backslash C, u\right).Pr\left(G_i\right)$$

where

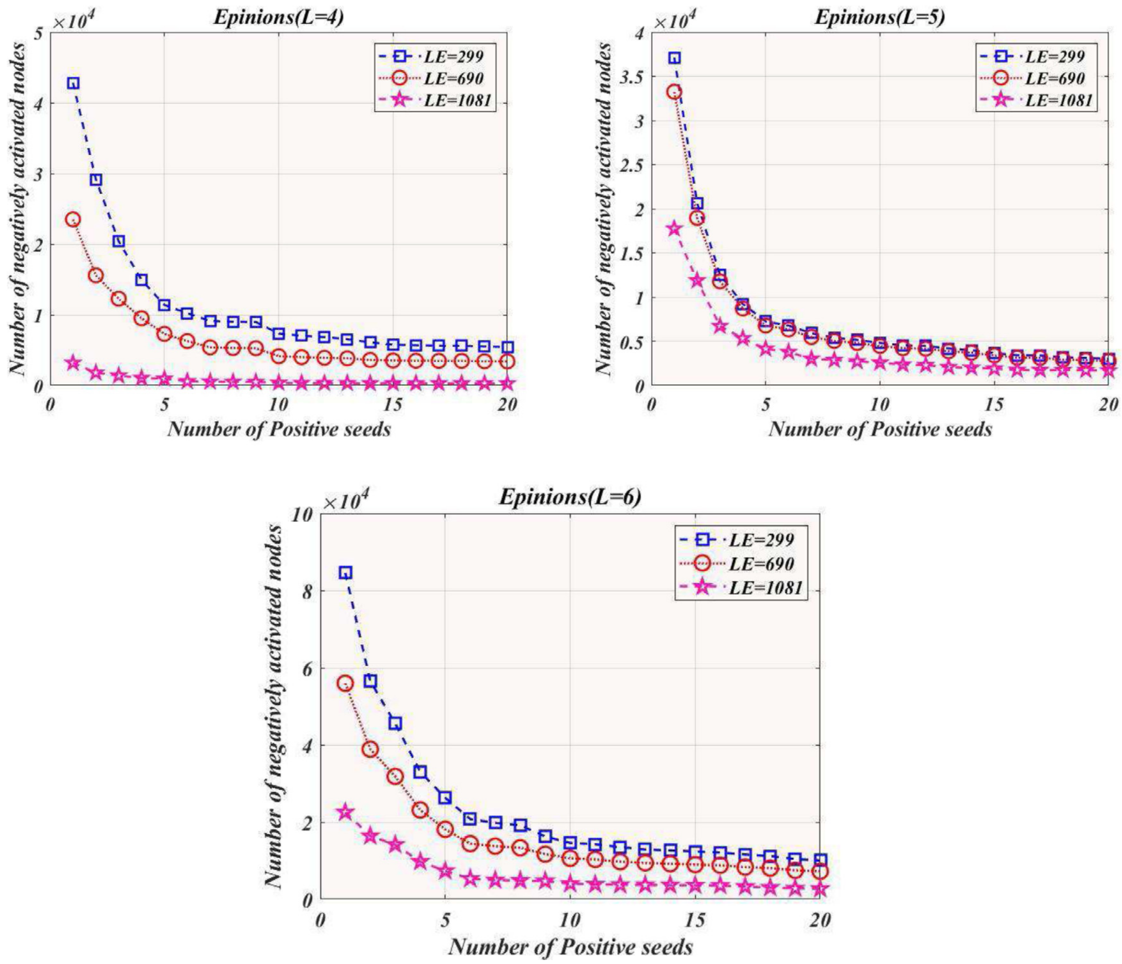

Fig. 15. Negative Influence Blocking on Epinions by different numbers of LE sub-graphs.

$\delta_{G_i}(V\backslash C, u) = \sum_{v\in V} I(v,u,G_i)$.

Therefore

$$\delta(V\backslash C,u) = \sum_{G_i\in W(G\backslash C)} Pr(G_i)\sum_{v\in V} I(v,u,G_i).$$

By (13), $I(u,v,G_i)$ is defined as

$$I(u,v,G_i)=\begin{cases}1 & \text{exists a route from } u \text{ to } v \text{ in } G_i\\ 0 & \text{otherwise}\end{cases}$$

Because $\sigma(V\backslash C,u)$ is a linear combination of $I(v,u,G_i)$, we only need to prove that $I(v,u,G_i)$ is super-modular with respect to C. Let G_i be a LE sub-graph generated in $G\backslash C$, and the corresponding LE sub-graph in graph $G\backslash(C\cup\{w\})$ be $G_i^{\overline{w}}$. For a node x in $V\backslash(C\cup\{w\})$, we denote the corresponding LE sub-graph in graph $G\backslash(C\cup\{x\})$ as $G_i^{\overline{x}}$. Similarly, we denote the corresponding LE sub-graph in graph $G\backslash(C\cup\{x,w\})$ as $G_i^{\overline{x,w}}$.

Then, we only need to prove:

$$I(v,u,G_i) - I\left(v,u,G_i^{\overline{w}}\right) \geq I\left(v,u,G_i^{\overline{x}}\right) - I(v,u,G_i^{\overline{x,w}}) \tag{24}$$

We consider the propagation tree $T_{G_i}^{v}$ rooted at v in G_i:

(1) If u is unreachable by v in the propagation tree $T_{G_i}^{v}$, the values of $I(v,u,G_i)$, $I\left(v,u,G_i^{\overline{w}}\right)$, $I\left(v,u,G_i^{\overline{x}}\right)$ and $I(v,u,G_i^{\overline{x,w}})$ are all 0, then $I(v,u,G_i) - I\left(v,u,G_i^{\overline{w}}\right) = I\left(v,u,G_i^{\overline{x}}\right) - I(v,u,G_i^{\overline{x,w}})$.

(2) If u is reachable by v in the propagation tree $T_{G_i}^{v}$, we denote the node set on the path from v to u in $T_{G_i}^{v}$ as $path(v,u)$. Then,

(2.1) If $w\in path(v,u)$ and $x\notin path(v,u)$, we have $I(v,u,G_i)=1$, $I\left(v,u,G_i^{\overline{w}}\right)=0$, $I\left(v,u,G_i^{\overline{x}}\right)=1$ and $I\left(v,u,G_i^{\overline{x,w}}\right)=0$, then $I(v,u,G_i) - I\left(v,u,G_i^{\overline{w}}\right) = I\left(v,u,G_i^{\overline{x}}\right) - I(v,u,G_i^{\overline{x,w}})$.

(2.2) If $w\notin path(v,u)$ and $x\in path(v,u)$, we have $I(v,u,G_i)=1$, $I\left(v,u,G_i^{\overline{w}}\right)=1$, $I\left(v,u,G_i^{\overline{x}}\right)=0$ and $I\left(v,u,G_i^{\overline{x,w}}\right)=0$, then $I(v,u,G_i) - I\left(v,u,G_i^{\overline{w}}\right) = I\left(v,u,G_i^{\overline{x}}\right) - I(v,u,G_i^{\overline{x,w}})$.

(2.3) If $w\notin path(v,u)$ and $x\notin path(v,u)$, we know $I(v,u,G_i)$, $I\left(v,u,G_i^{\overline{w}}\right)$, $I\left(v,u,G_i^{\overline{x}}\right)$ and $I\left(v,u,G_i^{\overline{x,w}}\right)$ are 1, then $I(v,u,G_i) - I\left(v,u,G_i^{\overline{w}}\right) = I\left(v,u,G_i^{\overline{x}}\right) - I(v,u,G_i^{\overline{x,w}})$.

(2.4) If $w\in path(v,u)$ and $x\in path(v,u)$, we know $I(v,u,G_i) = 1$, $I\left(v,u,G_i^{\overline{w}}\right)$, $I\left(v,u,G_i^{\overline{x}}\right)$ and $I\left(v,u,G_i^{\overline{x,w}}\right)$ are 0, then $I(v,u,G_i) - I\left(v,u,G_i^{\overline{w}}\right) > I\left(v,u,G_i^{\overline{x}}\right) - I(v,u,G_i^{\overline{x,w}})$.

Therefore, we have $I(v,u,G_i) - I\left(v,u,G_i^{\overline{w}}\right) \geq I\left(v,u,G_i^{\overline{x}}\right) - I(v,u,G_i^{\overline{x,w}})$ in all the cases, which indicates that $I(v,u,G_i)$ is super-modular. As a linear combination of $I(v,u,G_i)$, function $\delta(V\backslash C,u)$ is also super-modular with respect to C. □

Since $B(C,d) = \delta(V,d) - \delta(V\backslash C,d)$, and $\delta(V\backslash C,d)$ is super-modular with respect to C, $B(C,d)$ is sub-modular with respect to C. Noticing that the negative influence blocking function $B(C)$ is a linear combination of $B(C,d)$, $B(C)$ is sub-modular with respect to C.

5.2. Framework of the algorithm

Since the objective function of UNS-IBM problem is monotonic and sub-modular on the LE sub-graphs, we can use the greedy

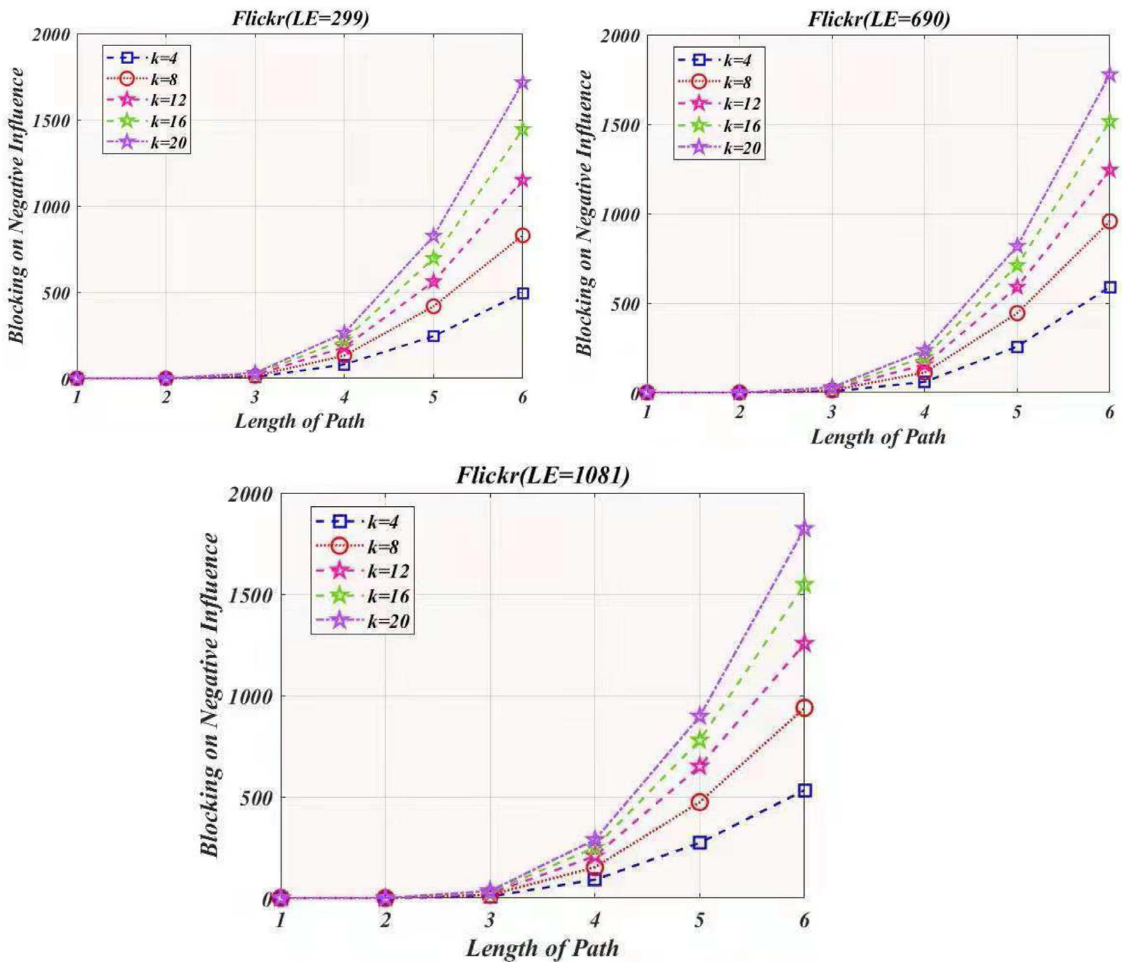

Fig. 16. Negative Influence Blocking by the algorithms on Flickr underdifferent lengths of the propagation paths.

approach to identify the optimal positive seed set. We present a node deletion-based algorithm *NDB* (node-deletion-blocking) for solving the UNS-IBM problem. The algorithm first constructs a set of LE sub-graphs G' and propagation trees $T^v_{G'}$ for each node v in G'. Then it calculates the initial values of $\Delta(w)$ and $r(v, T^v_{G'})$. In the algorithm, the positive seed set C is initialized as an empty one. It successively selects the node w with the highest blocking increment $\Delta(w)$ as the new positive seed. The procedure of seed selection will be repeated until $|C| = k$.

After a new positive seed v being selected and added to C, we need to update the values of $\Delta(w)$ andr $\left(v, T^v_{G'}\right)$ in LE sub-graph G'. Let C be the partial positive seed set obtained in the greedy approach. The blocking by C on the negative influence can be calculated in graph $G \backslash C$. After the new positive seed v joining C, the blocking by $C \cup \{v\}$ should be calculated in the graph $G \backslash (C \cup \{v\})$. Comparing each propagation tree $T^u_{G'_{V \backslash C}}$ in graph $G \backslash C$ with its counterpart $T^u_{G'_{V \backslash (C \cup \{v\})}}$ in graph $G \backslash (C \cup \{v\})$, the sub-tree rooted by v in $T^u_{G'_{V \backslash C}}$ is missing in $T^u_{G'_{V \backslash (C \cup \{v\})}}$. The difference of their coverages is $r\left(v, T^u_{G'}\right) + 1$. Based on such difference of coverages, the values of $r\left(w, T^u_{G'}\right)$ and $\Delta(w)$ of each node w in the sub-tree can be updated according to (18) and (19). Since node v is deleted in the updated LE sub-graph, the sub-tree rooted by v should also be deleted. The $r\left(w, T^u_{G'_{V \backslash C}}\right)$ value of node w in the sub-tree should be reset to 0. Meanwhile, the values of $r(w, T^u_{G'_{V \backslash C}})$ and $\Delta(w)$ of node v's ancestor w in the propagation tree $T^u_{G'_{V \backslash C}}$ should be updated accordingly. Framework of the algorithm *NDB* (node-deletion-blocking) is described in Fig. 4.

Computational complexity of Algorithm *NDB* is analyzed as follows. Let n and m respectively be the numbers of nodes and edges in $G = (V,E)$. Line 1 of Algorithm *NDB* calculates the sampling size R using (23) in O(1) time. Lines 2 to 5 construct LE sub-graphs by calling Algorithm *Construct_LE*. They also construct propagating trees and initialize $\Delta(u)$ for all the nodes by calling Algorithm *Construct_PT*. Since time complexities of Algorithms *Construct_LE* and *Construct_PT* are O(m) and O($m.n$) respectively, computations in lines 2 to 5 take O($R.m.n$) time. Based on the greedy approach, lines 7 to 30 select k positive seeds according to the blocking increment $\Delta(u)$. After node v being selected as the new positive seed, lines 10 to 28 update the values of $\Delta(w)$, $r\left(v, T^v_{G'}\right)$ for all the non-seed nodes in all the LE sub-graphs. To update these values for each node, a breadth-first-search is performed in O(m) time. Therefore, computations in lines 7 to 30 require O($R.k.m.n$) time. The total time for algorithm *NDB* is O($m.n$+$R.k.m.n$). Since R and k can be considered as constants, time complexity of algorithm *NDB* is O($m.n$).

6. Results

In order to evaluate the effectiveness of algorithm *NDB*, we conduct experiments in five real-world networks. In the experiments, we test the blocking effect of our algorithm *NDB,* and compare it with other influence blocking maximization algorithms.

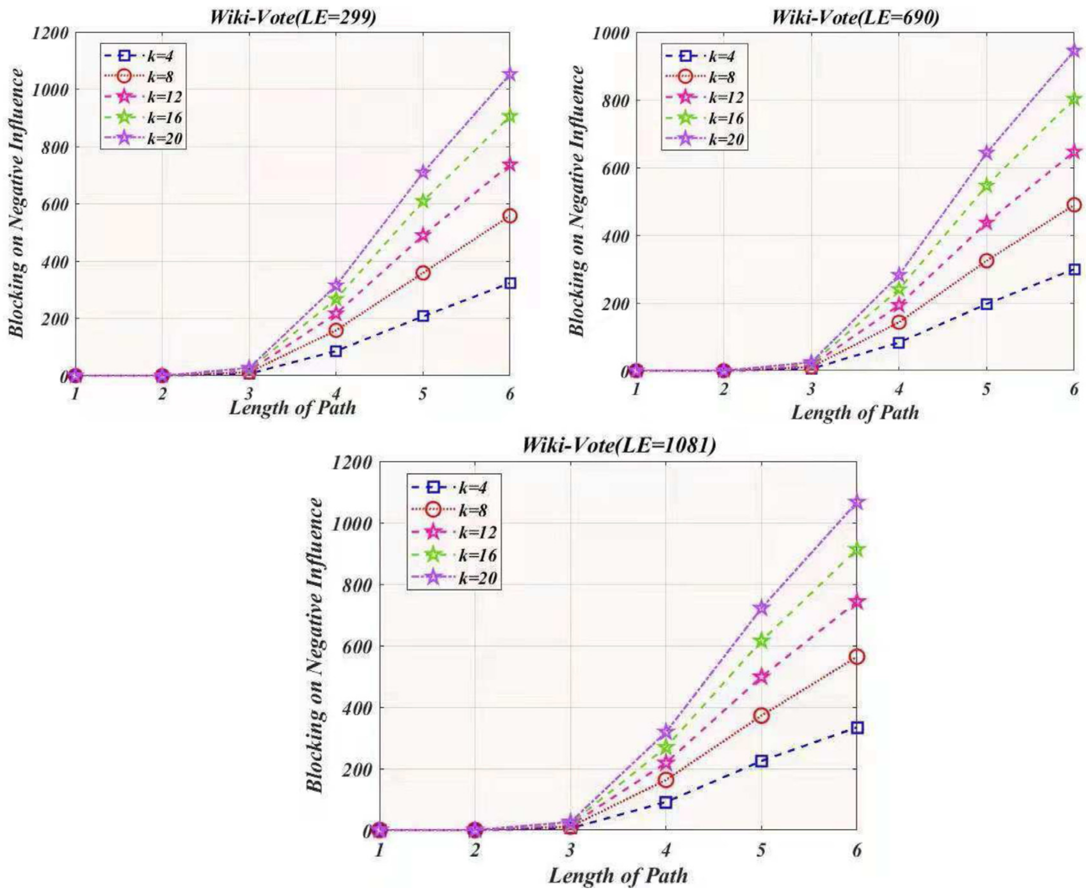

Fig. 17. Negative Influence Blocking by the algorithms on Wiki-Vote underdifferent lengths of the propagation paths.

In the tests, the algorithms are run on Intel (R) core (TM) i7-5700HQ CPU at 2.7G Hz and 8.0 GB RAM under Windows 10 system.

6.1. Data sets

In the experiments, the algorithms are tested on five real world datasets: Wiki-Vote, NetHEPT, NetPHY, Epinions, and Flickr.

Wiki-Vote [53]: This network is constructed based on Wikipedia and consists of voting data. In Wiki-Vote network, nodes stand for the individuals, and the directed edge from node u to v represents the vote of user u for v.

NetHEPT and **NetPHY** [54] These two datasets represent the academic research networks in the field of physics obtained from the website arXiv. The NetHEPT dataset consists of the authors and articles from the High Energy Physics-Theory section, while the NetPHY dataset consists of the authors and articles from Physics section.

Epinions [4]: This dataset represents a social network where users can discuss commodities on the market. Each user may post his viewpoints on the remarks by other users. Different opinions cause the opposing influence spreading in the network.

Flickr [55]: This network is constructed by adding edges between the images with common metadata in Flickr. In the network,each node stands for an image. There is an edge connecting two nodes if their images are from the same location, gallery, group, or set.

Table 2 shows the structural features of the networks tested, such as the numbers of edges and nodes, the minimum, maximum and average degrees.

Table 2 Structural features of five networks tested

Table 2
Structural features of five networks tested.

Datasets	#Node	#Edge	Average degree	Minimum degree	Maximum degree
Wiki-Vote	7115	103689	26.64	1	893
NetHEPT	15223	31387	7.75	1	44
NetPHY	37154	174161	12.48	1	131
Epinions	75879	508837	13.4	1	1801
Flickr	10879	39994	7.35	1	100

Fig. 5 shows the degree distributions of these networks tested.

In the experiments on these networks, we set the probability of influence spreading through each edge (u, v) as $1/d(v)$, where $d(v)$ denotes the degree of node v. The distribution p^- of the negative seeds is defined as $p^-(v) = e^{-\frac{1}{d(v)}}$ for each node v so that a node with higher degree will have a larger probability to be a negative seed.

6.2. Compared algorithms

In the experiments, the negative influence blocking by algorithm *NDB* is tested and compared with five other algorithms.

Random: This is a stochastic method that randomly chooses nodes as positive seeds. In the tests, the stochastic seed selection is performed for 1000 times, and the average blocking by the 1000 seed sets is the final result.

Degree [56]: The method treats the node with the highest degree centrality as the most influential one, and employs the heuristic method to select the most influential nodes as the positive seeds.

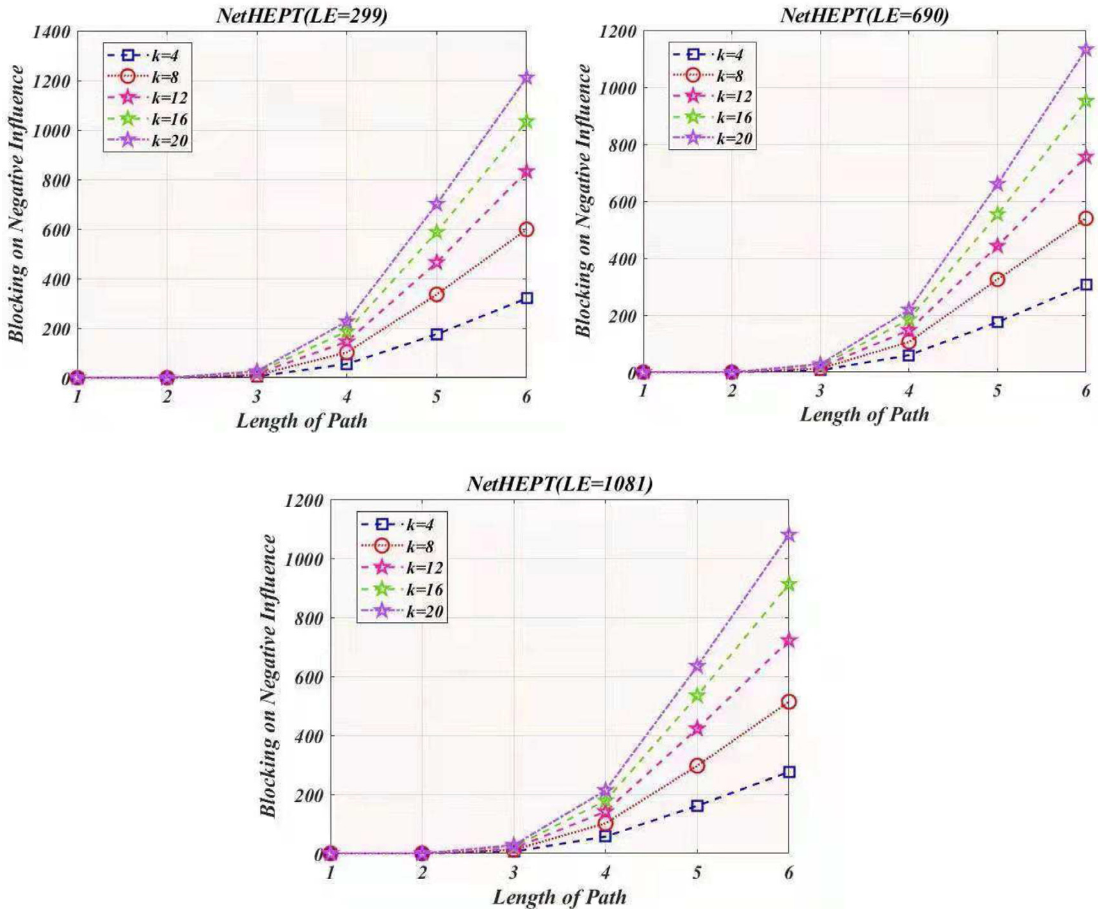

Fig. 18. Negative Influence Blocking by the algorithms on NetHEPT underdifferent lengths of the propagation paths.

Greedy [9]: The method first sets the positive seed set as an empty one, and then successively chooses the nodes which have the highest increments of negative influence blocking as the positive seeds. The increment of negative influence blocking by each node is estimated by the Monte Carlo sampling.

CLDAG [57]: This algorithm is designed under an extended LT model named CLT (competitive linear threshold). It utilizes the properties of the CLT model, and selects the positive seeds by computing the positive and negative activation probability based on a LDAG (local directed acyclic graph) structure.

ProxMinGreedy [40] This is a method proposed in 2020 for negative influence containment in social networks. The method uses an extended linear threshold model with one direction state transition (LT1DT) to describe the competitive information propagation in the network. The greedy approach is employed to minimize the false information spread in the social network.

We choose these IBM algorithms as the compared methods because they are suitable for the LT propagation model. In addition, these algorithms can be modified to solve the UNS-IBM problem on the uncertain sources by constructing a number of negative seed sets.

6.3. Testing on negative influence blockings by the algorithms

We test the algorithm *NDB* and compare its negative influence blocking with those of the compared algorithms. Since all the compared algorithms are for the IMB problem with determined sources, the positive seed selection strategies of these algorithms are marginally modified for solving the UNS-IBM with uncertain negative influence source. In the experiments on those algorithms, we construct 1000 negative seed sets D_i ($i = 1,2,\ldots,1000$), each of which consists of 200 randomly selected negative seeds based on the distribution p^-. Each algorithm is tested on all the negative seed sets D_i ($i = 1,2,\ldots,1000$). The average of the blocking effects on the influences by these negative seed sets is output as the final result.

We test the negative influence blockings of the methods under different positive seed sizes. In our experiments, we set the probability threshold as $\rho = 0.93$, and the error threshold as $\varepsilon = 0.05$. In each test, the size of the positive seed set changes from 0 to 20. Figs. 6 through 10 show and compare the numbers of negatively activated nodes by the algorithms tested on the datasets. The lower number of negatively activated nodes indicates the larger blocking effect of the results. From the figures, we can see that *NDB* achieves the highest blocking effect among all the methods. This demonstrates the effectiveness of algorithm *NDB*. The blockings on negative influence by *Random*, *Degree* and *Greedy* are very close to each other on most datasets, and are the lowest among all the methods. *CLDAG* yields better performance than *Random*, *Greedy* and *Degree* and exhibits moderate performance. *ProxMinGreedy* gets the second largest blocking next to NDB in most datasets.

The reason why *NDB* achieves the largest blocking on negative influence is that it uses a blocking function $B(C,d)$ to estimate the blocking effect by the positive seed set. Additionally, in order to get high precision results, *NDB* uses the propagation tree in the LE sub-graph to simulate the influence spreading, and uses node deletion to efficiently estimate the blocking effects by the positive seed sets.

It also can be observed from the figures that the blocking effects by *NDB*, *CLDAG* and *ProxMinGreedy* increase steadily following the expansion of the positive seed set. When the number

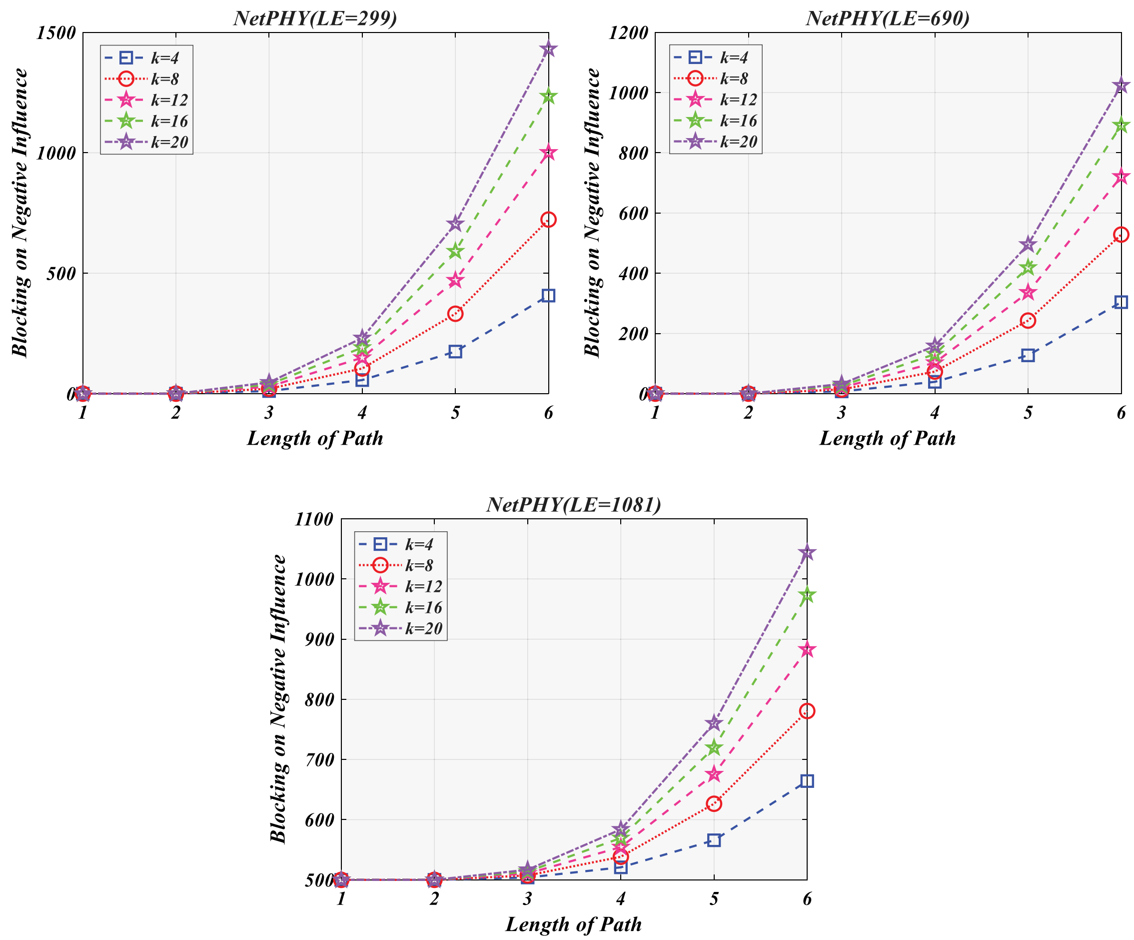

Fig. 19. Negative Influence Blocking by the algorithms on NetPHY under different lengths of the propagation paths.

of positive seeds is small, the blocking effect by *NDB* growths very fast. Following the expansion of the positive seed set, the increment of the blocking effect gradually slows down. This is because the influence spreading by different positive seeds may overlap with each other. More positive seeds may cause larger overlap of influence spreading, which makes the less increment of the blocking effects on negative influence.

The experiment results show that the negative influence blocking effect of algorithms *Degree* and *Random* are not changed much under different number of positive seeds. In the test on those two algorithms, when the number of positive seeds increases, less nodes will be negatively activated, but the decrease was not obvious. This is because the selection of positive seeds by these two algorithms does not depend on the information of negative seeds, but only on the degrees of nodes, or on a random selection. The influence range of the positive seeds selected does not aim at the range of negative influences. With more positive seeds, although the influence scope of positive seeds increases, the blocking effect on the negative influence does not increase significantly.

6.4. Influence blocking by different sampling sizes

To verify the effectiveness of the proposed *NDB* algorithm, we test the negative influence blockings with different numbers of the LE sub-graphs constructed. In this test, we set the probability threshold as $\rho = 0.93$. According to Theorem 1, if we set the error threshold as $\varepsilon = 0.1$, 0.08 and 0.05, we can get the number of the sampled LE sub-graphs as $R = 299$, 690 and 1081 respectively. In the test on each sampling size R, we change the number of the positive seeds from 0 to 20. In each test, the initial number of the positive seeds is assigned as $N = 0$, and its increment is set as $\Delta N = 2$. In the test on each dataset, we set the length limits of the propagation paths as $L = 4$, 5 and 6. The negative influence blockings under different sampling sizes on the five datasets are shown in Figs. 11 through 15.

From the figures, it can be observed that the blocking effect increases monotonically with the number of sampled LE graphs. When the sampling size is set as 1081, the algorithm gets the lowest negative influence spreading in all the datasets, which demonstrates the highest blocking effect on negative influence spreading. Conversely, when the sampling size is set as 299, the algorithm gets the lowest blocking effect on negative influence spreading. From the figures, we can see that with the small number of positive seeds, the sampling size has large effect of on the result. However, when the number of seeds increases, the difference of blocking effects under different sampling sizes is not significant. For instance, in the test on dataset Wiki-Vote with $L = 6$, the difference between the blocking effects by different sampling sizes is less than 100 when the positive seed size exceeds 15. But when the positive seed size is less than 15, their difference can be as large as 1000. This is due to the overlap of influence spreading by different positive seeds. When the number of positive seeds increases, larger overlap of influence spreading reduces the difference between the blocking effects by different sampling sizes. These results demonstrate that the *NDB* algorithm is not sensitive on the sampling sizes when the number of positive seeds is very large. In this case, a larger sampling size does not necessarily result in a higher quality result.

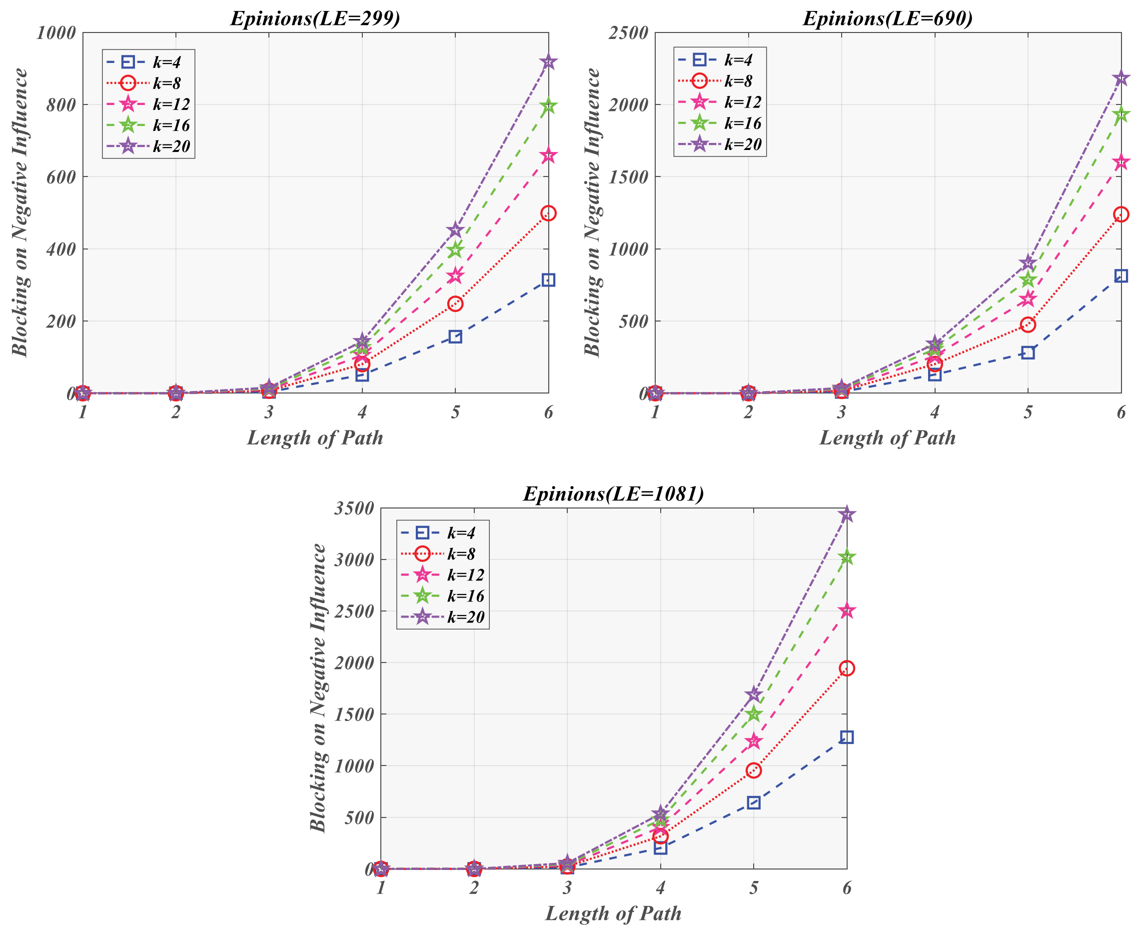

Fig. 20. Negative Influence Blocking by the algorithms on Epinions under different lengths of the propagation paths.

6.5. Influence blocking by different lengths of propagation paths

We also test and compare the influence blockings by algorithm *NDB* with different lengths L of propagation paths on the five datasets. In the experiments, we set the numbers of the sampled LE sub-graphs as 299, 690 and 1081, which are derived by setting the error thresholds as $\varepsilon = 0.1$, 0.08 and 0.05 respectively. The numbers of positive seeds are set as $k = 4$, 8, 12, 16 and 20. Figs. 16 through 20 show the influence blockings on the datasets under the different lengths of propagation paths and the different numbers of positive seeds. Such influence blockings are the reduction of the negatively activated nodes. From the figures, we can see that a longer propagation path results in larger blocking effect due to the wilder propagation range it covers.

6.6. Test on the computation times

We also test and compare the computation times consumed by the algorithms. In the experiments, we set the size of sampled LE sub-graphs as 299. The number of positive seeds is set as 200, and the length of propagation path is set as $L = 6$. Since all the compared algorithms are for the IMB with determined sources, the positive seed selection strategies of these algorithms are marginally modified for solving the UNS-IBM problem with uncertain negative influence source. In the experiments on those algorithms, they are tested on multiple negative seed sets D_i ($i = 1,2,\ldots,1000$) which are randomly generated according to the distribution p^-. The average of the blocking effects on the negative influences of D_i is output as the final result. Fig. 21 shows the computational times consumed by the algorithms on the datasets. It can be observed from the figure that *NDB* requires much less computation time than the other algorithms.

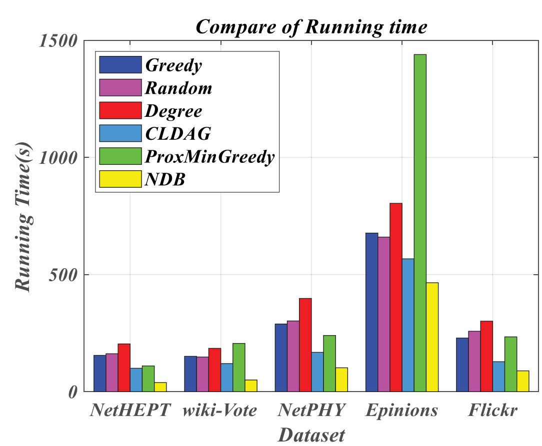

Fig. 21. Computation times of the algorithms.

The reason for the low computational complexity of algorithm *NDB* is that it uses the LE sub-graphs to approximate the influence spreading. Algorithm *NDB* can estimate the stochastic influence spreading in the determined LE sub-graphs. After constructing the propagation trees in the LE sub-graphs, *NDB* computes the influence blocking effects of the candidate seeds by estimating their coverages in the LE sub-graphs. Such node coverage can be calculated very fast in the propagation trees.

From the experimental results shown above, we can come to the conclusion that *NDB* can achieve more blocking on negative influence and requires less computational time than the other methods.

7. Discussion

For solving the uncertain negative source influence blocking maximization (UNS-IBM) problem, we define an extended LT propagation model CI-LTPM on the competitive social networks. Unlike the traditional LT model used in the other methods [15,32,40,57], the CI-LTPM propagation model can more accurately describe the simultaneous propagation of positive and negative influences in the network, while the traditional LT model only for one type of influence propagation.

Based on the CI-LTPM model, a node deletion-based method is proposed for solving the UNS-IBM problem. Our node deletion-based method is quite different from the IBM method which limits the negative influences by blocking some selected nodes [37]. The goal of our method is to find some seeds which can propagate positive influence to maximally counteract the negative influence. However, the goal of the method in [37] is to find and block the nodes close to the sources of the negative influence. This makes the method not applicable to the UNS-IBM problem where the negative sources are unknown.

The computational time of our method is much less than the other methods [9,40,57] which use the time-consuming simulation on influence spreading. The low time cost of our method is due to the propagation tree for estimating the influence propagation on the LE sub-graph. Particularly, we propose an algorithm to calculate the blocking increments of the positive seeds by node deletion on the propagation trees.

Compared with other methods for the traditional IBM problem where the negative influence sources are known in advance [17,18,21], our method can detect the positive seeds to block the negative influence from the unknown sources. The method estimates the probability of each LE sub-graph so that the blocking on the negative influences from the uncertain sources can be expressed by the expectation of the decrease of the negative influence on the LE sub-graphs.

We propose a method to estimate the number of LE sub-graphs. Experimental results show that based on the estimated number of the LE sub-graphs, our method can balance the computation time and the accuracy of the results. Abundant experimental results show that the proposed node-delete method can block more negative influence than the other methods. In addition, the computational time required is less than the other methods [9,40,57].

In our experiments, we found some results that are not observed by the other methods [18–20]. First, we found from the tests that when the number of positive seeds is small, the blocking effect growths very fast. Following the expansion of the positive seed set, the increment of the blocking effect gradually slows down. This is because the influence spreading by different positive seeds may overlap with each other. More positive seeds may cause larger overlap of influence spreading, which makes less increment of the blocking effects. Second, we observed that with the smaller number of positive seeds, the sampling size has larger effect on the result. However, when the number of positive seeds increases, the difference between the blocking effects under different sampling sizes is not significant. When the number of positive seeds increases, larger overlap of influence spreading reduces the increment of the blocking effects. In this case, a larger sampling size does not necessarily result in a proportionate reduction in the negative influence.

Although our method is based on the CI-LTPM propagation model which is an extension of the LT model [58], its basic ideas can be applied to other propagation models such as the independent cascade (IC) model. For solving the UNS-IBM problem under the IC model, we can use the sketch graph [58] instead of the LE sub-graph. Based on such sketch graph, the **reverse reachable** set (RRS) can be used to estimate the influence propagation. Similarly, the blocking increments of the positive seeds can be estimated based on node deletion on the propagation trees in the sketch graph.

8. Conclusion and future works

Due to the commercial and political competition in the society, positive and negative influences may spread in social networks simultaneously. The problem of negative influence blocking maximization (IBM) has aroused extensive interest of the researchers. In real world social networks, the exact source of negative influence is usually unknown. Instead, we only know the distribution of negative seeds. It is a challenge to detect the positive seed set to maximize the blocking on the negative influence from such uncertain source. In this work, we define the issue of uncertain negative source influence blocking maximization (UNS-IBM). We propose a competitive influence linear threshold propagation model (CI-LTPM) for the UNS-IBM problem. Based on the CI-LTPM model, we define the propagation tree on the LE sub-graph for estimating the influence propagation. An algorithm *Construct_PT* is proposed to construct the propagation tree and to calculate the blocking increment of each positive seed based on the propagation tree in the LE sub-graph. It can be observed that the blocking on negative influence by a positive seed is the reduction of the negative influence after the positive seed being deleted from the LE sub-graph. Based on such observation, we present a node deletion-based algorithm *NDB* for detecting the positive seeds to maximally block the negative influence from uncertain sources. Our experiment results demonstrate that algorithm *NDB* achieves more blocking on negative influence than other methods in less computational time.

In many social networks, the distribution of the negative seeds is likely changing over time. In this case, the positive seed identifying method must be modified correspondingly. One of our future works is to develop an efficient method to block the negative influence from dynamic sources. Recently, many social networks are capable of recording geographical location of the users. In our further research work, we will investigate the IBM and UNS-IBM problems in location-based networks, which aim to identify positive seed sets to maximally block the negative influence in a certain geographical area.

CRediT authorship contribution statement

Weijia Ju: Software (lead), Writing – original draft. **Ling Chen:** Conceptualization, Methodology (lead), Writing – review & editing (equal). **Bin Li:** Methodology (supporting), Validation (equal). **Yixin Chen:** Methodology (supporting), Writing – review & editing (equal). **Xiaobing Sun:** Software (supporting), Validation (equal), Formal analysis.

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 research was supported in part by the Chinese National Natural Science Foundation under grant Nos. 61972335, 61872312, 61379066, 61702441, 61472344, 61402395, 61906100 and 61602202; Natural Science Foundation of Jiangsu Province under contracts BK20201430, Jiangsu Provincial Graduate Education Innovation Project (KYCX19_2097)

References

[1] N.M. Ahmed, L. Chen, Yulong Wang, Bin Li, Yun Li, Wei Liu, Deep-Eye: Link prediction in dynamic networks based on non-negative matrix factorization, Big Data Min. Anal. 1 (1) (2018) 19–33.

[2] Y.P. Chen, Semantics-aware influence maximization in social networks, Inform. Sci. 513 (2020) 442–464, http://dx.doi.org/10.1016/j.ins.2019.10.075.

[3] M. Hosseini-Pozveh, K. Zamanifar, A.R. Naghsh-Nilchi, Assessing information diffusion models for influence maximization in signed social networks, Expert Syst. Appl. 119 (1) (2019) 476–490.

[4] J. Leskovec, Epinions social network, 2019, http://snap.stanford.edu/data/soc-Epinions1.html.

[5] L. Yang, Z.W. Li, A. Giua, Containment of rumor spread in complex social networks, Inf. Sci. 506 (2020) 113–130, http://dx.doi.org/10.1016/j.ins.2019.07.055.

[6] H.M. Huang, Recurrent neural variational model for follower-based influence maximization, Inf. Sci. 528 (2020) 280–293, http://dx.doi.org/10.1016/j.ins.2020.04.023.

[7] Z. Wang, W. Dong, W. Zhang, C.W. Tan, Rumor source detection with multiple observations: Fundamental limits and algorithms, ACM SIGMETRICS Perform. Eval. Rev. 42 (1) (2014) 1–13.

[8] D. D. Li, C. H.Wang, S.P. Zhang, G.L. Zhou, D.H. Chu, C. C. Wu, Positive influence maximization in signed social networks based on simulated annealing, Neurocomputing 260 (2017) 69–78.

[9] C.V. Pham, H.M. Dinh, H.D. Nguyen, H.T. Dang, H.X. Hoang, Limiting the spread of epidemics within time constraint on online social networks, in: Proceedings of the Eighth International Symposium on Information and Communication Technology, SoICT 2017, 2017, pp. 262–269.

[10] C. Gao, Z. Su, J.M. Liu, J. Kurths, Even central users do not always drive information diffusion, Commun. ACM 62 (2) (2019) 61–67.

[11] K. Tanınmış, N. Aras, I.K. Altınel, Influence maximization with deactivation in social networks, European J. Oper. Res. 278 (1) (2019) 105–119.

[12] W.J. Ju, L. Chen, B. Li, W. Liu, J. Sheng, Y.W. Wang, A new algorithm for positive influence maximization in signed networks, Inform. Sci. 512 (2020) 1571–1591, http://dx.doi.org/10.1016/j.ins.2019.10.061.

[13] A. Bondielli, F. Marcelloni, A survey on fake news and rumor detection techniques, Inf. Sci. 497 (2019) 38–55, http://dx.doi.org/10.1016/j.ins.2019.05.035.

[14] J. Ratkiewicz, M. Conover, M. Meiss, B. Gonçalves, A. Flammini, F. Menczer, Detecting and tracking political abuse in social media, in: Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media, 2011.

[15] Patryk Pazura, Jarosław Jankowski, Kamil Bortko, Modeling the performance of limiting information spreading processes under competing linear threshold model, Procedia Comput. Sci. 176 (2020) 3751–3760, http://dx.doi.org/10.1016/j.procs.2020.09.012.

[16] W.Y. Liu, K. Yue, H. Wu, et al., Containment of competitive influence spread in social networks, Knowl.-Based Syst. 109 (2016) 266–275, http://dx.doi.org/10.1016/j.knosys.2016.07.008.

[17] P. Wu, L. Pan, Scalable influence blocking maximization in social networks under competitive independent cascade models, Comput. Netw. 123 (2017) 38–50.

[18] J.G. Lv, B. Yang, Z. Yang, W. Zhang, A community-based algorithm for influence blocking maximization in social networks, Cluster Comput. 22 (2019) 5587–5602.

[19] Q.H. Shi, C. Wang, D.S. Ye, J.W. Chen, Y. Feng, C. Chen, Adaptive influence blocking: minimizing the negative spread by observation-based policies, in: Proceedings of 2019 IEEE 35th International Conference on Data Engineering, ICDE, 2019, pp. 1502–1513.

[20] C. Yao, Y.X. Zhang, X.M. Zhang, K.G. Bian, L.Y. Song, Competitive influence blocking in online social networks: A case study on wechat, in: Proceedings of 24th Asia-Pacific Conference on Communications, APCC, 2018, pp. 251–256.

[21] W.L. Zhu, W. Yang, S.C. Xuan, D.P. Man, W. Wang, X.J. Du, M. Guizani, Location-based seeds selection for influence blocking maximization in social networks, IEEE Access 7 (2019) 27272–27287.

[22] A. Şimsek A, R. Kara, Using swarm intelligence algorithms to detect influential individuals for influence maximization in social networks, Expert Syst. Appl. 114 (2018) 224–236.

[23] D. Li, W. Wang, J.M. Liu, Grassroots VS elites: Which ones are better candidates for influence maximization in social networks? Neurocomputing 358 (17) (2019) 321–331.

[24] A. CaliÒ, Attribute based diversification of seeds for targeted influence maximization, Inform. Sci. (2020) http://dx.doi.org/10.1016/j.ins.2020.08.093.

[25] L.F. Li, Targeted influence maximization under a multifactor-based information propagation model, Inform. Sci. 519 (2020) 124–140, http://dx.doi.org/10.1016/j.ins.2020.01.040.

[26] Y.J. Yang, Y.B. Xu, E. Wang, K.H. Lou, Dongming Luan, Exploring influence maximization in online and offline double-layer propagation scheme, Inform. Sci. 450 (2018) 182–199, http://dx.doi.org/10.1016/j.ins.2018.03.048.

[27] S. Singh, K. K. Singh, A. Kumar, B. Biswas, MIM2: Multiple influence maximization across multiple social networks, Physica A 526 (2019) Article 120902.

[28] F. Wang, J.H. She, Y. Ohyama, W.J. Jiang, Geyong Min, G.J. Wang, M. Wu, Maximizing positive influence in competitive social networks: A trust-based solution, Inf. Sci. 546 (6) (2021) 559–572, http://dx.doi.org/10.1016/j.ins.2020.09.002.

[29] F. Wang, W.J. Jiang, X.L. Li, G.J. Wang, Maximizing positive influence spread in online social networks via fluid dynamics, Future Gener. Comput. Syst. 86 (2018) 1491–1502.

[30] D.D. Yang, X.W. Liao, H.W. Shen, X.Q. Cheng, G.L. Chen, Relative influence maximization in competitive social networks, SCI. CHINA Inf. Sci. 60 (10) (2017) 108101.

[31] M.M.D. Khomami, A. Rezvanian, N. Bagherpour, M.R. Meybodi, Minimum positive influence dominating set and its application in influence maximization: a learning automata approach, Appl. Intell. 48 (3) (2018) 570–593.

[32] A. Bozorgi, S. Samet, J. Kwisthout, T. Wareham, Community-based influence maximization in social networks under a competitive linear threshold model, Knowl.-Based Syst. 134 (15) (2017) 149–158.

[33] C. Budak, D. Agrawal, A. El Abbadi, Limiting the spread of misinformation in social networks, in: Proceedings of the 20th International Conference on World Wide Web, 2011, pp. 665–674.

[34] X. Deng, F. Long, B. Li, D. Cao, Y. Pan, An influence model based on heterogeneous online social network for influence maximization, IEEE Trans. Netw. Sci. Eng. 7 (2) (2020) 737–749.

[35] J. Jankowski, P. Bartkow, P. Pazura, K. Bortko, Evaluation of the costs of delayed campaigns for limiting the spread of negative content, panic and rumours in complex networks, in: V. Krzhizhanovskaya, et al. (Eds.), ICCS 2020, in: Lecture Notes in Computer Science, vol. 12140, Springer, Cham, 2020, http://dx.doi.org/10.1007/978-3-030-50423-6_22.

[36] N. N. Arazkhani, M.R. Meybodi, An efficient algorithm for influence blocking maximization based on community detection, in: 2019 5th International Conference on Web Research, ICWR, 2019, pp. 258–263, http://dx.doi.org/10.1109/ICWR.2019.8765277.

[37] S. Wang, X. Zhao, Y. Chen, Z. Li, K. Zhang, J. Xia, Negative influence minimizing by blocking nodes in social networks, in: AAAIWS'13-17: Proceedings of the 17th AAAI Conference on Late-Breaking Developments in the Field of Artificial Intelligence, 2013, pp. 134–136.

[38] Q. Yao, C. Zhou, L. Xiang, Y. Cao, L. Guo, Minimizing the negative influence by blocking links in social networks, in: International Conference on Trustworthy Computing and Services, ISCTCS 2014, Trustworthy Computing and Services, pp. 65-73.

[39] H. Zhang, H. Zhang, X. Li, M.T. Thai, Limiting the spread of misinformation while effectively raising awareness in social networks, in: International Conference on Computational Social Networks, CSoNet 2015, Computational Social Networks, pp. 35-47.

[40] L. Yang, Z. Li, A. Giua, Containment of rumor spread in complex social networks, Inform. Sci. 506 (2020) 113–130.

[41] J. Archbold, N. Griffiths, Limiting concept spread in environments with interacting concepts, in: AAMAS '17: Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems, 2017, pp. 1332–1340.

[42] J. Tsai, T.H. Nguyen, N. Weller, et al., Game-theoretic target selection in contagion-based domains, Comput. J. 57 (6) (2014) 893–905, http://dx.doi.org/10.1093/comjnl/bxt094.

[43] J. Li, K. Yue, D.H. Zhang, et al., Robust influence blocking maximization in social networks, J. Comput. Res. Dev. 53 (3) (2016) 601–610, http://dx.doi.org/10.7544/issn1000-1239.2016.20148341.

[44] G. Jiang, S. Li, M. Li, Dynamic rumor spreading of public opinion reversal on Weibo based on a two-stage SPNR model, Physica A 558 (2020) 125005, http://dx.doi.org/10.1016/j.physa.2020.125005.

[45] L.A. Huo, J.H. Jiang, S.X. Gong, B. He, Dynamical behavior of a rumor transmission model with Holling-type II functional response in emergency event, Physica A 45015 (2016) 228–240.

[46] G.H. Chen, Ilscr rumor spreading model to discuss the control of rumor spreading in emergency, Physica A 52215 (2019) 88–97.

[47] X.F. Dong, Y.J. Liu, C. Wu, Y. Lian, D.S. Tang, A double-identity rumor spreading model, Physica A 52815 (2019) Article 121479.

[48] 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. 66 (3) (2017) 2789–2800.

[49] A.I.E. Hosni, K. Li, S. Ahmad, Minimizing rumor influence in multiplex online social networks based on human individual and social behaviors, Inform. Sci. 512 (2020) 1458–1480.

[50] A.I.E. Hosni, K. Li, Minimizing the influence of rumors during breaking news events in online social networks, Knowl.-Based Syst. 1936 (2020) 105452.

[51] L.H. Zhu, B.X. Wang, Stability analysis of a SAIR rumor spreading model with control strategies in online social networks, Inform. Sci. 526 (2020) 1–19.
[52] J. Guo, Y. Li, W. Wu, Targeted protection maximization in social networks, IEEE Trans. Netw. Sci. Eng. 7 (3) (2020) 1645–1655.
[53] J. Leskovec, Wikipedia vote network, 2019, http://snap.stanford.edu/data/wiki-Vote.html.
[54] http://www.arXiv.org.
[55] https://snap.stanford.edu/data/web-flickr.html.
[56] C.V. Pham, D.K. Ha, D.Q. Ngo, Q.g C. Vu, H.X. Hoang, A new viral marketing strategy with the competition in the large-scale online social networks, in: Proceedings of the IEEE RIVF International Conference on Computing & Communication Technologies, Research, Innovation, and Vision for the Future, 2016, pp. 1–6.
[57] X. He, G. Song, W. Chen., et al., Influence blocking maximization in social networks under the competitive linear threshold model, in: Proceedings of the 2012 SIAM International Conference on Data Mining, pp 463–474.
[58] C. Borgs, M. Brautbar, J. Chayes, B. Lucier, Maximizing social influence in nearly optimal time, in: Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms, SODA '14, 2014, pp. 946–957.