Maximizing Boosted Influence Spread with Edge Addition in Online Social Networks

LEI YU, GUOHUI LI, and LING YUAN, Huazhong University of Science and Technology, China

Influence maximization with application to viral marketing is a well-studied problem of finding a small number of influential users in a social network to maximize the spread of influence under certain influence cascade models. However, almost all previous studies have focused on node-level mining, where they consider identifying nodes as the initial seeders to achieve the desired outcomes. In this article, instead of targeting nodes, we investigate a new boosted influence maximization problem from the edge-level perspective, which asks for finding an edge set that is added to the network to maximize the increased influence spread of a given seed set. We show that the problem is NP-hard and the influence spread function no longer exhibits the property of submodularity, which impose more challenging on the problem. Therefore, we devise a restricted form that is submodular and propose a greedy algorithm with approximate guarantee to solve the problem. However, because of its poor computational efficiency, we further propose an improved greedy algorithm that integrates several effective optimization strategies to significantly speed up the edge selection without sacrificing its accuracy. Extensive experiments over real-world available social networks of different sizes demonstrate the effectiveness and efficiency of the proposed methods.

CCS Concepts: • **Information systems** → **Data mining**; *Web applications*; • **Theory of computation** → Design and analysis of algorithms;

Additional Key Words and Phrases: Boosted influence maximization, viral marketing, online social networks, edge addition

ACM Reference format:
Lei Yu, Guohui Li, and Ling Yuan. 2020. Maximizing Boosted Influence Spread with Edge Addition in Online Social Networks. *ACM/IMS Trans. Data Sci.* 1, 2, Article 9 (May 2020), 21 pages.
https://doi.org/10.1145/3364993

1 INTRODUCTION

The increasing prosperity of online social networks such as Facebook, Myspace, and Twitter provide more convenient platforms for mutual communication and information dissemination. As a cost-effective marketing strategy on these online social networks, viral marketing [4] aims to provide price-discounted or even free samples of a product to a group of the influential users in a social network, and make the most of the powerful word-of-mouth effect [9, 19], the influence propagation can eventually trigger a large number of product adoptions. Motivated by the viral marketing strategy, the influence maximization problem [14, 23, 37] has become one of the

Authors' address: L. Yu and L. Yuan (corresponding author), School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, Hubei, China; emails: {LYU91, cherryyuanling}@hust.edu.cn; G. Li, School of Software Engineering, Huazhong University of Science and Technology, Wuhan 430074, Hubei, China; email: guohuili@hust.edu.cn.

2577-3224/2020/05-ART9 $15.00
https://doi.org/10.1145/3364993

most fundamental problems in this field and has also attracted much research attention in both theoretical and practical.

Formally, the seminal work by Kempe et al. [23] formulate influence maximization as a discrete optimization problem. The problem is defined as follows. A social network is modeled as a directed graph, where nodes represent users and directed edges reflect the relationships between different users. For a given budget K, the objective of the influence maximization problem is to identify at most K nodes to be activated initially, such that the expected number of nodes that are activated by them in the graph is maximized under certain influence cascade models when the influence propagation process ends. Kempe et al. propose two widely used influence propagation models that are denoted the Independent Cascade (IC) model and Linear Threshold (LT) model, which are both taken from mathematical sociology. In general, the IC model tends to emphasis on individual influence among friends. While the LT model tends to focus on the influence of group behaviors in the influence diffusion. Follow these two models, some other variant models have also been proposed such as triggering models [23], voter models [29, 42], and Susceptible/Infected/Susceptible (SIS) models [32].

Although large amounts of the approximate algorithms and heuristics [3, 6, 7, 21, 33, 39, 40] have been actively studied to solve the influence maximization problem effectively in social networks, almost all of these work only focuses on node-level seeders mining, i.e., they mainly consider maximizing one's expected influence spread or blocking the influence spread of one's competitors by nurturing a small number of the initial adopters (i.e., the seed nodes). However, in the actual marketing campaigns, there is also such a requirement for companies that to make the most of the limited marketing budget, they usually apply a mixture of multiple marketing strategies to promote their products. In other words, when the initial adopters are determined, the companies are also very interested in offering some additional incentives or rewards (e.g., coupon, small gift, etc.) to create some new connections among users. They hope that it is able to contribute to further increase the product adoption in the networks, which is directly related to earning more revenue by the viral marketing effort. Moreover, it is generally considered that the cost of creating new connections among users (e.g., the average redemption and distribution cost of the coupon or small gift) for the companies may be much smaller than the cost of selecting the initial adopters (e.g., the average cost of providing free products) in reality.

In fact, increasing new connections among users to facilitate the information dissemination is not scarce in real life. To better illustrate it, let us consider the following several real scenarios. To enlarge the influence of advertising among users, the advertisers may pay the providers of the social networking services for connecting web users. Some online social networking sites (e.g., Twitter, Weibo) offer a special function, where users are able to make new connections and become friends with other users based on the similarity of their profiles, interests, topic discussion community, and so on. Additionally, bookstores usually give small flavors such as pretty bookmark or membership card to earn more customers who do not know more about the bookstores before. The bookstores' owners hope that it can be more beneficial to sell their books and even widely spread their fames in the future.

Therefore, inspired by these more practical demands, we are interested in a problem of adding a limited number of new connections among users to further increase the influence spread in a network when the initial adopters are given. Unfortunately, very little previous work has taken into account such a problem. In addition to the initial adopters, it also provides the companies with more alternative marketing strategies and more flexible budget allocations to trigger substantial information spread under a limited budget, the problem can be considered as an important complement to the studies of the traditional influence maximization problems actually. However, it still has several aspects that are very different from them. Instead of targeting the initial seed nodes, the problem mainly focuses on the edges that are added to the network to further increase the influence spread of the seed nodes. Furthermore, as opposed to the influence maximization problems on a static network, the network structure in this problem has changed because of those new added edges, which imposes new challenges on the problem.

In practice, there may be many potential applications for the study of the above problem from the edge-level perspective. The link recommendation task, which is one of the useful applications, aims to suggesting a set of links to a user of the social network to greatly increase the social circle and connectivity of the user. As a result, it is possible to help the user to make more friends with the same hobbies or interests, and widely disseminate the innovation or idea. In addition, other promising applications include the target advertising. For example, an advertiser desires to promote a service to a high-value potential customer. However, because of the social distance between them, it may be useless to advertise the target customer directly. Therefore, it may be a good idea for the advertiser to first create connections with some friends of the target customer, which is more likely to increase the chances of promoting to the target customer.

In this article, we investigate a new Boosted Influence Maximization (BIM) problem from the edge-level prospective in social networks, which is a novel generalization of influence maximization. The BIM problem is to find an edge set of size at most N[1] that is added to the network to maximize the increased influence spread of a given seed set under the independent cascade model. We show that the BIM problem is NP-hard. However, as a result of the network structure changes with the edge addition, the influence spread function is monotone but not submodular. It means that the simple greedy algorithm may not be directly applied to the BIM problem. To tackle this challenge, we devise a restricted form that has the property of submodularity and is very close to the original influence spread in practice. Therefore, it allows for the greedy algorithm with approximate guarantee to solve the problem effectively. Due to the poor computational efficiency of the greedy algorithm in selecting the edges, we further propose an improved greedy algorithm that integrates several effective optimization strategies to filter out many unpromising candidate edges and avoid traversing each candidate edge to evaluate the incremental influence spread of the seed set in each iteration. With these optimization strategies, we are able to significantly speed up the edge selection while does not affect its accuracy.

To summarize, the main contributions of this article are as follows.

- We study a novel BIM problem from the edge-level prospective in social networks, which is very different from most of influence maximization problems. The problem is NP-hard, and computing the increased influence spread of a given seed set is #P-hard. Moreover, the influence spread function maintains the monotonicity but lacks the submodularity.
- We devise a restricted form of the influence spread function, and propose a greedy algorithm with approximate guarantee to solve the BIM problem effectively. Moreover, to overcome the poor computational efficiency, we propose an improved greedy algorithm to greatly accelerate selecting the edges, and devise an efficient heuristic method to approximate the increased influence spread.
- We evaluate the performance of the proposed methods over real-world available social networks of different scales and structural features, the experimental results demonstrate the effectiveness and efficiency of the proposed methods.

The rest of this article is structured as follows. Section 2 reviews the related work about influence maximization. In Section 3, we introduce the influence propagation model and give the definition

[1] Due to the limited marketing budget in practice, the N is limited and usually not very large after the seed nodes are selected.

of the BIM problem. Then, we show several challenges we face on the BIM problem in Section 4. Section 5 proposes several approximate algorithms. Section 6 presents the experimental results and analysis. Finally, we conclude this article and discuss several future directions in Section 7.

2 RELATED WORK

2.1 Influence Maximization

Domingos and Richardson [14, 37] were the first to study the influence maximization problem from an algorithmic perspective. They model the influence propagation process as Markov random fields and devise heuristic algorithms. Then, Kempe et al. [23] were the first to formulate influence maximization as a discrete optimization problem. Kempe et al. prove that the problem under the proposed influence propagation models is NP-hard. According to the properties of monotonicity and submodularity of the influence spread function, Kempe et al. develop a greedy hill-climbing algorithm, which can obtain an approximate solution with a lower bound ratio of $(1 - 1/e)$ to the optimal solution. However, there are two intrinsic drawbacks for the greedy algorithm. First, it needs to equally traverse all candidate nodes before selecting a new seed in each iteration. In addition, since computing the influence spread of a given seed set is #P-hard, Monte Carlo simulation–based method needs to run sufficient times to get an accurate estimate.

Therefore, a large number of algorithms have been proposed to improve the greedy algorithm. Based on the lazy procedure strategy, Leskovec et al. [28] propose an efficient CELF method, which can achieve about 700 times improvement on the greedy algorithm. The CELF method makes the best use of the submodularity to greatly reduce the number of the objective function calls. As an extension of CELF method, Goyal et al. [20] develop a CELF++ method. The CELF++ method can further improve the computational efficiency, which is reported to be about 35%–55% faster than CELF method. Motivated by the existence of community in a network, several work [8, 36, 41] have focused on employing the communities of the network, and proposing more efficient algorithms to speed up the seed selection. Additionally, instead of Monte Carlo simulation–based method, various heuristic algorithms have also been proposed to improve the efficiency in evaluating the influence spread for a given seed set in the greedy algorithm. By constructing a local region in which the influence flows, Chen et al. propose two effective and efficient heuristic algorithms of PMIA [6] and LDAG [7] to estimate the influence spread under the independent cascade model and linear threshold model, respectively. Both algorithms are able to achieve almost the same influence spread as the greedy algorithm empirically while far less time consumption. Jung et al. [21] propose a scalable IRIE method, which formulates the influence propagation process with linear equations. They efficiently estimate the incremental influence spread of each node in an iterative way. In recent work, Borgs et al. [3] propose a near-linear time algorithm based on reverse reachability search from the sampled nodes. The method can provide a $(1 - 1/e - \varepsilon)$ approximate solution with at least $(1 - n^{-l})$ probability. Tang et al. propose TIM method [40] and the improved IMM method [39] under the triggering model. These two methods can obtain the same approximate guarantee as the method proposed by Borgs et al. while achieving much higher empirical efficiency. Nguyen et al. [33] design two novel sampling algorithms called SSA and D-SSA. These two methods can also provide $(1 - 1/e - \varepsilon)$ approximate guarantee while achieving up to 1,200 times faster than the state-of-the-art IMM method. It means that both SSA and D-SSA methods have better scalability to efficiently handle very large social networks.

However, all of these work just focus on the node-level influence maximization that makes great effort to seek for the optimal seed nodes in a network. Different from them, we mainly study the edge-level boosted influence maximization, which concentrates on adding a limited number of new edges to further increase the influence spread when the initial seed nodes are given.

2.2 Network Modification Problem

The network modification problem aims to optimize certain network properties by adding or removing a limited number of edges in a network. For edge addition, Ghosh et al. [18] study the problem that adds a set of new edges into a graph to maximize the algebraic connectivity measuring how well-connected the graph is. In Reference [13], Reference [35], and Reference [15], the authors study the problem of minimizing the diameter of a graph by adding k shortcut edges. Papagelis et al. [34] consider adding some edges to minimize the characteristic path length of a graph. They propose a novel path screening sampling-based method to speed up the edge selection. Crescenzi et al. [10] focus on the problem of adding k edges into both directed and undirected graphs to maximize the closeness of a pre-defined node. Meanwhile, several work about edge removal have also been explored. In References [24–26], the authors focus on the problem that minimizes the diffusion of undesirable things (e.g., computer virus, malicious rumor, etc.) by blocking some edges. Kuhlman et al. [27] consider the contagion blocking problem in networked populations, which aims at identifying some edges to be removed under a simpler deterministic variant of the LT model. However, although these work also focus primarily on the analysis of edge-level in a network, they have never involved influence propagation and maximization.

The most similar to our work is Reference [1] and Reference [11]. However, these two work is simple and has a severe limitation that they just recommend links that can only be incident to the initial seed nodes, which is not very common in the practical marketing applications. More importantly, the goals they studied are also totally different from ours.

3 INFLUENCE PROPAGATION MODEL AND PROBLEM DEFINITION

In this section, we first introduce the basic independent cascade model, which describes how information is spread in a social network. Then, we present the definition of the BIM problem. Table 1 lists the notations to be extensively used in the rest of this article.

3.1 Independent Cascade Model

Independent Cascade (IC) model, which is one of the basic influence propagation models, is popularly used in the influence maximization problems. For a given directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, each edge (u, v) in $\mathcal{E}$ is associated with a non-negative weight function $\omega(u, v)$: $\mathcal{E} \rightarrow [0, 1]$, which represents the probability that node v is successfully activated by u through the edge (u, v) after node u becomes active. In particular, if $(u, v) \notin \mathcal{E}$, it satisfies $\omega(u, v) = 0$. Each node in the IC model has two states, which are either active or non-active. Moreover, the state of each node can switch from non-active to active, but not vice verse. Generally, the IC model works as follows. At the time step 0, a seed set $\mathcal{S} \subseteq \mathcal{V}$ is targeted and becomes active initially, while all other nodes are non-active. The influence propagation process proceeds in the discrete time steps $t = 0, 1, 2, \ldots$. Let $\mathcal{S}_t$ be a set of activated nodes at the time step $t (t \geq 0)$, and it satisfies $\mathcal{S}_0 = \mathcal{S}$. At the time step $t + 1$, each active node u in $\mathcal{S}_t$ makes attempt to independently activate each of its currently non-active neighbors $v \in \mathcal{V} \setminus \cup_{0 \leq i \leq t} \mathcal{S}_i$ with an activation probability $\omega(u, v)$. Once the node v is activated successfully, it stays active and continues to activate its non-active neighbors similar to the above process in the next time step. It is worth noting that each node can only be activated at most once in the IC model. The influence propagation terminates at a time step when there is no more nodes to be activated.

3.2 Problem Definition

For the directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, we can formally define the BIM problem under the IC model as follows. Given two positive integers N and $K (K < |\mathcal{V}|)$, a candidate edge set C and a seed

Table 1. Notation Explanation

Notation	Description
$\mathcal{G} = (\mathcal{V}, \mathcal{E})$	A directed graph with node set $\mathcal{V}$ and edge set $\mathcal{E}$
$\omega(u, v)$	The activation probability on each edge (u, v) in $\mathcal{E}$
$\mathcal{R}$	A set of edges to be added to $\mathcal{G}$
C	A set of candidate edges not in $\mathcal{E}$
N	The number of edges in $\mathcal{R}$
K	The number of nodes in the given seed set $\mathcal{S}$
$\mathcal{I}(\mathcal{S};\mathcal{R})$	The number of activated nodes with the seed set $\mathcal{S}$ after adding a new edge set $\mathcal{R}$ to $\mathcal{G}$
$\Delta\mathcal{I}(\mathcal{S};\mathcal{R})$	The increased influence spread of the seed set $\mathcal{S}$ when a new edge set $\mathcal{R}$ is added to $\mathcal{G}$
$Pr(\mathcal{P}_{u,v})$	The probability of node v is activated by u through the path $\mathcal{P}_{u,v}$ in $\mathcal{G}$
$Pr(\mathcal{S}, u; \mathcal{R})$	The probability of node u is activated by the seed set $\mathcal{S}$ after adding a new edge set $\mathcal{R}$ to $\mathcal{G}$
γ	The submodularity ratio of a monotone set function that belongs to $[0, 1]$
$\Delta\mathcal{I}(e\|\mathcal{S};\mathcal{R})$	The incremental influence spread of the seed set $\mathcal{S}$ with the edge set $\mathcal{R}$ when adding a new edge e to $\mathcal{G}$
$\Delta\mathcal{I}_m$	The maximum incremental influence spread of the seed set $\mathcal{S}$ with an edge set in the current iteration in $\mathcal{G}$

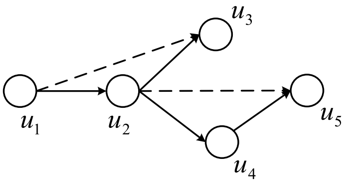


Fig. 1. The example of a directed graph for the BIM problem, and the dotted lines represent the added edges.

set $\mathcal{S}$ of size K where $\mathcal{S} \subseteq \mathcal{V}$. The activation probability on each edge $(u, v) \in \mathcal{E}$ is defined as $\omega(u, v)$. Let $\mathcal{I}(\cdot)\colon 2^{\mathcal{V}} \to \mathbb{R}$ be a set function such that $\mathcal{I}(\mathcal{S};\mathcal{R})$ is the expected number of activated nodes with the seed set $\mathcal{S}$ after adding a new edge set $\mathcal{R}$ to the graph $\mathcal{G}$ under the IC model. Let $\Delta\mathcal{I}(\mathcal{S};\mathcal{R})$ represent the increased influence spread of $\mathcal{S}$ when the edge set $\mathcal{R}$ is added, which equals $\mathcal{I}(\mathcal{S};\mathcal{R}) - \mathcal{I}(\mathcal{S})$. Therefore, the goal of BIM problem is to find an edge set $\mathcal{R}^*$ of size at most N from C to maximize the increased influence spread $\Delta\mathcal{I}(\mathcal{S};\mathcal{R})$. The BIM problem can be expressed as follows:

$$\mathcal{R}^* = argmax_{\mathcal{R} \subseteq C, |\mathcal{R}| \leq N} \Delta\mathcal{I}(\mathcal{S};\mathcal{R}). \tag{1}$$

We take the following example to better illustrate the BIM problem. Consider a directed graph with nodes $u_1, u_2, \ldots, u_5$ shown in Figure 1. Let each edge have an activation probability 0.5, and let the set C be $\{(u_1, u_3), (u_2, u_5), (u_4, u_3), (u_5, u_3)\}$, where we select two edges that are added to the graph. We assume that the size of the seed set $\mathcal{S}$ is 1. Therefore, the node u_1 should be selected as the seed (i.e., $\mathcal{S} = \{u_1\}$) in the graph due to it achieves the largest influence spread under the IC model. Specially, when $\mathcal{R} = \emptyset$, which means that there is no edge to be added to the graph, it satisfies $\mathcal{I}(\mathcal{S}) = 2.125$ and $\Delta\mathcal{I}(\mathcal{S};\mathcal{R}) = 0$. When adding the edges (u_1, u_3) and

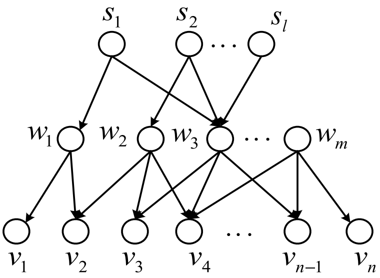


Fig. 2. A directed graph with $n + m + l$ nodes.

(u_2, u_5), it has $\Delta\mathcal{I}(\mathcal{S}; \{(u_1, u_3), (u_2, u_5)\}) = 0.594$, which achieves the maximal increased influence spread given the seed set. Therefore, the optimal solution for the BIM problem is the edge set $\mathcal{R} = \{(u_1, u_3), (u_2, u_5)\}$. Meanwhile, we can see that when the edge set $\mathcal{R}$ is added to the graph, the influence spread of $\mathcal{S}$ has increased by about 28% than before. It demonstrates that the BIM problem focusing on the edge-level is very different from the traditional influence maximization problems actually, and the added edges are able to significantly increase the influence spread of the seed set in the network.

4 CHALLENGES OF THE PROPOSED PROBLEM

In this section, we present several challenges we face on the BIM problem. Additionally, we also analyze the properties of the influence spread function in social networks.

Similar to the traditional influence maximization problem, the BIM problem under the IC model is also NP-hard, which is presented in the following theorem.

Theorem 4.1. *The BIM problem is NP-hard under the IC model.*

Proof. First, let us remind the classical set cover problem. For a given ground set $\mathcal{U} = \{u_1, u_2, \ldots, u_n\}$, the collection of subsets of the set $\mathcal{U}$ is defined as $\mathcal{T} = \{t_1, t_2, \ldots, t_m\}$. The set cover problem aims to find whether there are k subsets in the set $\mathcal{T}$ such that the union of these subsets is the set $\mathcal{U}$. It assumes $k < n < m$. Then, we prove that the set cover problem can be viewed as a special case of the BIM problem as follows.

For an arbitrary instance of the set cover problem, we define a directed graph $\mathcal{G}'$ with $n + m + l$ nodes, which is shown in Figure 2. Specifically, for the graph $\mathcal{G}'$ in Figure 2, there is a seed set $\mathcal{S}$ that contains l nodes, i.e., $\mathcal{S} = \{s_1, s_2, \ldots, s_l\}$. A set of nodes $\mathcal{W} = \{w_1, w_2, \ldots, w_m\}$ contains m nodes, where each node w_i corresponds to a subset t_i in $\mathcal{T}$. A set of nodes $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ contains n nodes, where each node v_i corresponds to an element u_i in $\mathcal{U}$. Furthermore, there are some directed edges from the seed nodes in $\mathcal{S}$ to the nodes in $\mathcal{W}$ where the activation probability on each edge is 1, and a directed edge from node w_i to v_i with activation probability 1 if a subset t_i in $\mathcal{T}$ contains the element u_i in $\mathcal{U}$. When no edge is added, the influence spread of $\mathcal{S}$ in $\mathcal{G}'$ is calculated as $\mathcal{I}(\mathcal{S}) = l + q + |\cup_{1 \le j \le q} t_j|$, where $|\cup_{1 \le j \le q} t_j| < n$ and $q (q < m)$ represents the number of nodes to be activated by $\mathcal{S}$ in $\mathcal{W}$ and it assumes that the activated nodes are $w_1, w_2, \ldots, w_q$. Specially, we choose k nodes from $\mathcal{W}$. Therefore, the set cover problem is equivalent to deciding if there is an edge set $\mathcal{R}$ that is added to the graph $\mathcal{G}'$, and the edges point from $\mathcal{S}$ to those k nodes or from those k nodes to $\mathcal{V}$. In this situation, the influence spread of $\mathcal{S}$ with the added edge set $\mathcal{R}$ satisfies $\mathcal{I}(\mathcal{S}; \mathcal{R}) = \mathcal{I}(\mathcal{S}) + \Delta\mathcal{I}(\mathcal{S}; \mathcal{R}) \ge l + k + n$, where $k \ge q$. In fact, it implies that the set cover problem can be reduced to an instance of the BIM problem. Due to the set cover problem is NP-complete [22], we can get that the BIM problem is also NP-hard. □

To solve the BIM problem in Equation (1), it has to calculate the increased influence spread $\Delta\mathcal{I}(\mathcal{S};\mathcal{R})$ of the seed set $\mathcal{S}$ with a new added edge set $\mathcal{R}$ under the IC model. However, the calculation of the increased influence spread $\Delta\mathcal{I}(\mathcal{S};\mathcal{R})$ is #P-hard, which is presented in the following theorem.

Theorem 4.2. *Computing the increased influence spread of the seed set $\mathcal{S}$ with a new added edge set $\mathcal{R}$ is #P-hard under the IC model.*

Proof. The calculation of the influence spread $\mathcal{I}(\mathcal{S};\mathcal{R})$ can be regarded as a reduction from an instance of the classical counting problem of $s-t$ connectness in a directed graph, which has been reported in Reference [6]. Because it has been shown in Reference [16] that the $s-t$ connectness problem is #P-complete, it can get that the calculation of the increased influence spread $\Delta\mathcal{I}(\mathcal{S};\mathcal{R})$ under the IC model is also #P-hard. □

According to the aforementioned Theorem 4.2, the #P-hard of computing the increased influence spread $\Delta\mathcal{I}(\mathcal{S};\mathcal{R})$ is one of the challenges on the BIM problem. Therefore, it needs a method to approximate the increased influence spread in a graph. To achieve better performance in computing $\Delta\mathcal{I}(\mathcal{S};\mathcal{R})$, we apply the extended Maximum Influence Arborescence (MIA) model [6]. More specifically, for any two nodes u and v in $\mathcal{V}$, a path from node u to v is denoted $\mathcal{P}_{u,v} = (u = u_1, u_2, \ldots, u_m = v)$, where there is no duplicate nodes. The probability that node v is activated by u through the path $\mathcal{P}_{u,v}$ is calculated as $Pr(\mathcal{P}_{u,v}) = \prod_{i=1}^{m-1} \omega(u_i, u_{i+1})$. It means that all nodes along the path need to be successfully activated. Since there may be multiple paths between nodes u and v, it only chooses the path with the maximum influence probability that is called maximal influence path in the MIA model, i.e., $MIP(u,v) = argmax_{\mathcal{P}\in\mathcal{P}(u,v|\mathcal{G})}\{Pr(\mathcal{P})\}$, where $\mathcal{P}(u,v|\mathcal{G})$ represents all paths between nodes u and v in the graph $\mathcal{G}$. Therefore, node u can activate v only through $MIP(u,v)$ in the MIA model. Moreover, to more efficiently compute the increased influence spread within the tolerance of error, it also uses an influence threshold θ to filter out the insignificant maximal influence paths (i.e., $Pr(MIP(u,v)) < \theta$) due to they have a very small impact on the influence spread computation. Specially, for a given edge set $\mathcal{R}$, we can calculate the influence spread $\mathcal{I}(\mathcal{S};\mathcal{R})$ as $\sum_{u\in\mathcal{V}} Pr(\mathcal{S},u;\mathcal{R})$, where $Pr(\mathcal{S},u;\mathcal{R})$ represents the probability that node u can be activated by the seed set $\mathcal{S}$ when the edge set $\mathcal{R}$ is added. Under the MIA model, it applies the maximum influence in(out) arborescence, which includes all significant maximal influence paths to(from) node u, to effectively and efficiently estimate the influence spread. Let $\mathcal{N}(u)$ be the neighbor nodes of u. When node $u \in \mathcal{S}$, it has $Pr(\mathcal{S},u;\mathcal{R}) = 1$. Otherwise $Pr(\mathcal{S},u;\mathcal{R})$ can be calculated as $1 - \prod_{w\in\mathcal{N}(u)}(1 - Pr(\mathcal{S},w;\mathcal{R})Pr(w,u;\mathcal{R}))$. Accordingly, we can finally calculate the influence spread $\mathcal{I}(\mathcal{S};\mathcal{R})$ as follows:

$$\begin{aligned}\mathcal{I}(\mathcal{S};\mathcal{R}) &= \sum_{u\in\mathcal{V}} Pr(\mathcal{S},u;\mathcal{R}) \\ &= \sum_{u\in\mathcal{V}} \left(1 - \prod_{w\in\mathcal{N}(u)} (1 - Pr(\mathcal{S},w;\mathcal{R})Pr(w,u;\mathcal{R}))\right).\end{aligned} \tag{2}$$

Similarly, the probability $Pr(\mathcal{S},w;\mathcal{R})$ can be recursively calculated based on the neighbor nodes of w. According to Equation (2), we are able to approximately compute the increased influence spread $\Delta\mathcal{I}(\mathcal{S};\mathcal{R})$. Let $n = |\mathcal{V}|$ be the number of nodes in the graph $\mathcal{G}$. For any node $u \in \mathcal{V}$, we can compute the maximum influence in arborescence to the node u for each seed in $\mathcal{S}$ by applying the Dijkstra shortest path algorithm. Let t_{in} represent the maximum running time to compute the arborescence for any u when an edge set $\mathcal{R}$ is added to $\mathcal{G}$. Therefore, from Equation (2), we can see that the time complexity of calculating $\Delta\mathcal{I}(\mathcal{S};\mathcal{R})$ is $O(n(t_{in} + m_{in}))$, where m_{in} denotes the average number of edges in the arborescences to other nodes in $\mathcal{V}$ for the seed set $\mathcal{S}$.

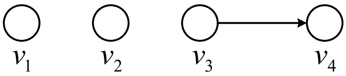


Fig. 3. The example of a directed graph with four nodes.

For any two sets $\mathcal{S}_1$ and $\mathcal{S}_2$ where $\mathcal{S}_1 \subseteq \mathcal{S}_2 \subseteq \mathcal{V}$, a set function $\mathcal{F}: 2^{\mathcal{V}} \rightarrow \mathbb{R}$ is monotone if $\mathcal{F}(\mathcal{S}_1) \leq \mathcal{F}(\mathcal{S}_2)$. Meanwhile, for the two sets $\mathcal{S}_1 \subseteq \mathcal{S}_2 \subseteq \mathcal{V}$ and any node $w \in \mathcal{V} \setminus \mathcal{S}_2$, the set function $\mathcal{F}$ is submodular if $\mathcal{F}(\mathcal{S}_1 \cup \{w\}) - \mathcal{F}(\mathcal{S}_1) \geq \mathcal{F}(\mathcal{S}_2 \cup \{w\}) - \mathcal{F}(\mathcal{S}_2)$. The submodularity [30, 31] captures the law of diminishing marginal returns, a well-known principle in economics. In particular, the influence spread function in the BIM problem has the property of monotonicity, which is presented in the following theorem.

Theorem 4.3. *The influence spread function $\Delta\mathcal{I}(\mathcal{S};\mathcal{R})$ is monotone under the IC model.*

Proof. For a given edge set $\mathcal{R}$ and the seed set $\mathcal{S}$, it is not hard to find that $\mathcal{I}(\mathcal{S};\mathcal{R}) \leq \mathcal{I}(\mathcal{S};\mathcal{R} \cup \{e\})$ when adding any new edge e to the graph $\mathcal{G}$. Therefore, it has $\Delta\mathcal{I}(\mathcal{S};\mathcal{R}) \leq \Delta\mathcal{I}(\mathcal{S};\mathcal{R} \cup \{e\})$. According to the definition of monotonicity of a set function, it can imply that the influence spread function $\Delta\mathcal{I}(\mathcal{S};\mathcal{R})$ is monotone. □

However, unlike most of influence maximization problems, the influence spread function in the BIM problem is no longer submodular. To better understand this property, we give a counter example as follows. Consider a directed graph with four nodes and one directed edge, which is shown in Figure 3, we assume that the activation probability on each edge is 0.2, the candidate edge set C is $\{(v_1, v_2), (v_2, v_3), (v_1, v_3)\}$ and the seed set $\mathcal{S}$ is $\{v_1\}$. When no edge is added, the increased influence spread $\Delta\mathcal{I}(\mathcal{S};\mathcal{R})$ is 0 due to the influence spread of $\mathcal{S}$ does not change. When adding the edges (v_1, v_2) and (v_2, v_3), respectively, the increased influence spread $\Delta\mathcal{I}(\mathcal{S};\{(v_1, v_2)\}) = 0.2$ and $\Delta\mathcal{I}(\mathcal{S};\{(v_2, v_3)\}) = 0$. Meanwhile, when both edges (v_1, v_2) and (v_2, v_3) are added, it has $\Delta\mathcal{I}(\mathcal{S};\{(v_1, v_2), (v_2, v_3)\}) = 0.248$. In particular, if $\mathcal{R}_1 = \emptyset$, $\mathcal{R}_2 = \{(v_1, v_2)\}$ and a new added edge e is (v_2, v_3), then we can calculate $\Delta\mathcal{I}(\mathcal{S};\mathcal{R}_1 \cup \{e\}) - \Delta\mathcal{I}(\mathcal{S};\mathcal{R}_1) = 0$ and $\Delta\mathcal{I}(\mathcal{S};\mathcal{R}_2 \cup \{e\}) - \Delta\mathcal{I}(\mathcal{S};\mathcal{R}_2) = 0.248 - 0.2 = 0.048$ based on Equation (2). Therefore, it implies that $\Delta\mathcal{I}(\mathcal{S};\mathcal{R}_1 \cup \{e\}) - \Delta\mathcal{I}(\mathcal{S};\mathcal{R}_1) < \Delta\mathcal{I}(\mathcal{S};\mathcal{R}_2 \cup \{e\}) - \Delta\mathcal{I}(\mathcal{S};\mathcal{R}_2)$ for $\mathcal{R}_1 \subseteq \mathcal{R}_2$. According to the above analysis, we have the following theorem.

Theorem 4.4. *The influence spread function $\Delta\mathcal{I}(\mathcal{S};\mathcal{R})$ is not submodular under the IC model.*

Therefore, the non-submodularity of the influence spread function imposes another challenge on the BIM problem.

5 APPROXIMATE ALGORITHMS

To tackle the challenge of the non-submodularity in the BIM problem, we devise an approximate influence spread function and first propose a greedy algorithm with approximate guarantee to solve the problem. Due to the greedy algorithm is very inefficient and expensive in the edge selection, we then propose an improved greedy algorithm that integrates several optimization strategies. This algorithm is able to significantly accelerate selecting the edges while does not affect the solution quality.

5.1 Greedy Algorithm

Given a monotone set function $\mathcal{F}$, we define its submodularity ratio [12] as follows.

Definition 5.1. Given two disjoint sets $\mathcal{S}$ and $\mathcal{T}$ that satisfy $\mathcal{S}, \mathcal{T} \subseteq \mathcal{V}$, for any node $v \in \mathcal{T}$, the submodularity ratio of the monotone set function $\mathcal{F}$ is the largest scalar $\gamma \in [0, 1]$ such

ALGORITHM 1: Simple Greedy Algorithm

```
Input: $\mathcal{G} = (\mathcal{V}, \mathcal{E}), \mathcal{S}, C, N$.
Output: $\mathcal{R}$.
Initialize: $\mathcal{R} \leftarrow \emptyset$;
Calculate the activation probability on each edge $(u, v)$ with $\omega(u, v) = 1/d(v)$;
for $i \leftarrow 1$ to $N$ do
    for each edge $e \in C$ do
        $\Delta\mathcal{I}(e|\mathcal{S};\mathcal{R}) \leftarrow \Delta\mathcal{I}(\mathcal{S};\mathcal{R} \cup \{e\}) - \Delta\mathcal{I}(\mathcal{S};\mathcal{R})$;
    end
    $e_m \leftarrow argmax\{\Delta\mathcal{I}(e|\mathcal{S};\mathcal{R})|e \in C\}$;
    $\mathcal{R} \leftarrow \mathcal{R} \cup \{e_m\}$;
    $\mathcal{E} \leftarrow \mathcal{E} \cup \{e_m\}$;
    $C \leftarrow C \setminus \{e_m\}$;
end
return $\mathcal{R}$.
```

that $\frac{\sum_{v \in \mathcal{T}} \mathcal{F}(v|\mathcal{S})}{\mathcal{F}(\mathcal{T}|\mathcal{S})} \geq \gamma$, where $\mathcal{F}(v|\mathcal{S})$ is the marginal gain by adding the node v into $\mathcal{S}$, i.e., $\mathcal{F}(v|\mathcal{S}) = \mathcal{F}(\mathcal{S} \cup \{v\}) - \mathcal{F}(\mathcal{S})$. Specially, the monotone set function $\mathcal{F}$ is submodular if and only if its submodularity ratio γ is 1.

For the submodularity ratio of the monotone set function $\mathcal{F}$, we have the following important lemma [2].

Lemma 5.2. *For the monotone set function $\mathcal{F}$ with submodularity ratio $\gamma \in [0, 1]$, the greedy algorithm that runs l steps can return a set $\mathcal{S}_l$ of size l such that $\mathcal{F}(\mathcal{S}_l) \geq (1 - e^{-\gamma \frac{l}{k}})\mathcal{F}(opt_k)$, where opt_k represents the optimal set of size k where $k \geq l$.*

It can imply from the aforementioned Lemma 5.2 that the approximate solution generated by the greedy algorithm may arbitrary bad when the objective function is only monotone but not submodular, which mainly relies on its submodularity ratio. For the BIM problem, it has been shown previously that the influence spread function is monotone and non-submodular. Therefore, to allow for the greedy algorithm, we devise a restricted form of the influence spread function that has the submodularity to approximate the calculation of the increased influence spread of the seed set with a new added edge set in a graph. We first define the restricted maximal influence path between different nodes as follows.

Definition 5.3. In the graph $\mathcal{G}$, if a maximal influence path from a seed node in $\mathcal{S}$ to another node u in $\mathcal{V} \setminus \mathcal{S}$ only includes at most one new added edge from C, it declares that the path is a restricted maximal influence path between the seed node and node u.

According to the above Definition 5.3, when computing the increased influence spread $\Delta\mathcal{I}(\mathcal{S};\mathcal{R})$ of the seed set $\mathcal{S}$ with a new added edge set $\mathcal{R}$ based on Equation (2), we instead consider that the influence flows only along the restricted maximal influence paths from each seed in $\mathcal{S}$ to any node u in the graph $\mathcal{G}$ and these paths are independent of each other in the arborescences. In this case, the original influence spread function is called a restricted influence spread function accordingly, which exhibits the desirable submodularity [5] and can approximate the original influence spread in practice. Specially, the time complexity of computing $\Delta\mathcal{I}(\mathcal{S};\mathcal{R})$ is $O(nt'_{in})$ in this case, where t'_{in} denotes the maximum running time to compute the maximum influence in arborescence for any node in which it only finds the restricted maximal influence paths among nodes when using the Dijkstra shortest path algorithm. Let us examine Figure 3 again. For the seed set $\mathcal{S} = \{v_1\}$, it is still true that the increased influence spread $\Delta\mathcal{I}(\mathcal{S};\{(v_2, v_3)\}) = 0$ and $\Delta\mathcal{I}(\mathcal{S};\{(v_1, v_2)\}) = 0.2$ when the edges (v_2, v_3) and (v_1, v_2) are added, respectively. However, when both edges (v_2, v_3) and (v_1, v_2) are added, it holds that $\Delta\mathcal{I}(\mathcal{S};\{(v_1, v_2), (v_2, v_3)\}) = 0.2$. It is different from the former case, because only the restricted maximal influence paths from the seed node v_1 to other nodes are considered. Therefore, we can calculate $\Delta\mathcal{I}(\mathcal{S};\mathcal{R}_1 \cup \{e\}) - \Delta\mathcal{I}(\mathcal{S};\mathcal{R}_1) = 0$ and $\Delta\mathcal{I}(\mathcal{S};\mathcal{R}_2 \cup \{e\}) - \Delta\mathcal{I}(\mathcal{S};\mathcal{R}_2) = 0$, which means that the restricted influence spread function is submodular. In practice, the restricted influence spread is very close to the original one in evaluating the increased influence spread for a new added edge set in a network, which has also been reported in Reference [5]. On the one hand, for the real world social networks, the number of edges to be added from C is much smaller than the number of edges in the networks actually. On the other hand, the maximal influence paths among nodes have the greatest opportunity for the flow of influence. Moreover, the longer paths are more likely to include more than one new added edge than the shorter paths, and tend to have smaller influence probabilities in turn. As a result, these paths have very little impact on the calculation of the increased influence spread in reality. Therefore, we are able to propose a greedy algorithm to effectively solve the BIM problem.

Formally, the pseudocode of the greedy algorithm is presented in Algorithm 1. In each iteration, the algorithm selects the edge with the largest incremental influence spread of the seed set $\mathcal{S}$ from the candidate edge set C, and adds it into the edge set $\mathcal{R}$. When the number of edges in $\mathcal{R}$ is N, the algorithm terminates and returns the final edge set $\mathcal{R}$. The time complexity of Algorithm 1 is $O(N|C|\mathcal{T}(\Delta\mathcal{I}(\mathcal{S};\mathcal{R})))$, where $|C|$ is the number of edges in C and $\mathcal{T}(\Delta\mathcal{I}(\mathcal{S};\mathcal{R}))$ represents the time for computing the increased influence spread $\Delta\mathcal{I}(\mathcal{S};\mathcal{R})$ in the graph $\mathcal{G}$. Additionally, due to the restricted influence spread function is monotone and submodular, it means that its submodularity ratio γ is 1 based on Definition 5.1. Therefore, according to Lemma 5.2, the greedy algorithm presented in Algorithm 1 can approximate the optimal solution with a lower bound ratio of $1 - 1/e$ for the BIM problem.

5.2 Improved Greedy Algorithm

However, when the size of the graph and the number of edges in the candidate edge set are both large, the greedy algorithm is very inefficient due to it has to equally traverse each candidate edge to calculate the incremental influence spread of the seed set in each iteration. Therefore, we further propose an improved greedy algorithm, which can significantly speed up the edge selection.

First, the greedy algorithm is required to check each candidate edge to evaluate the increased influence spread of the seed set $\mathcal{S}$ in each iteration. However, it may be some unpromising candidate edges that do not need to be exactly checked. In other words, there may be no path from the seed nodes in $\mathcal{S}$ to the start nodes of certain candidate edges. As a result, the seed set $\mathcal{S}$ is impossible to activate the start nodes and their following nodes when those candidate edges are added to the graph. Therefore, the increased influence spread of the seed set $\mathcal{S}$ will not change. For this situation, we filter out those unpromising candidate edges to narrow the space of the set C in each iteration. Therefore, we have the following lemma.

Lemma 5.4. *For a candidate edge (u, v), it can be filtered out if there is no path from the seed nodes in $\mathcal{S}$ to the node u.*

We are able to apply the efficient Breadth-First Search method to find the unpromising candidate edges from the set C in the graph. Moreover, its computational cost is generally smaller than the cost of exactly computing the increased influence spread of the seed set $\mathcal{S}$ with the new added edges, especially when the graph is relatively large.

In addition, the greedy algorithm must repeatedly evaluate the incremental influence spread of the seed set $\mathcal{S}$ for all candidate edges in the graph $\mathcal{G}$. However, its computational complexity may

ALGORITHM 2: Improved Greedy Algorithm

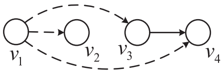
```
Input: $\mathcal{G} = (\mathcal{V}, \mathcal{E}), \mathcal{S}, C, N$.
Output: $\mathcal{R}$.
Initialize queue $\mathcal{M}_i (i = 0, 1, \ldots, N)$;
Initialize: $\mathcal{R}_0 \leftarrow \emptyset$, $\mathcal{M}_0 \leftarrow \emptyset$;
Calculate the activation probability on each edge $(u, v)$ with $\omega(u, v) = 1/d(v)$;
for $i \leftarrow 1$ to $N$ do
    $\Delta\mathcal{I}_m \leftarrow 0$;
    for each edge $e = (u, v) \in C$ do
        if $u$ is not reachable from $\mathcal{S}$ then
            continue; //based on Lemma 5.4
        end
        if $e \in \mathcal{M}_{i-1}$ and $\Delta\mathcal{I}(\mathcal{S}; \mathcal{R}_{i-2} \cup \{e\}) - \Delta\mathcal{I}(\mathcal{S}; \mathcal{R}_{i-2}) < \Delta\mathcal{I}_m$ then
            continue; // based on Lemma 5.5
        else
            Calculate the incremental influence spread of $\mathcal{S}$ with the edge $e$,
            $\Delta\mathcal{I}(e|\mathcal{S}; \mathcal{R}_{i-1}) \leftarrow \Delta\mathcal{I}(\mathcal{S}; \mathcal{R}_{i-1} \cup \{e\}) - \Delta\mathcal{I}(\mathcal{S}; \mathcal{R}_{i-1})$, and insert $(e, \Delta\mathcal{I}(e|\mathcal{S}; \mathcal{R}_{i-1}))$ into $\mathcal{M}_i$;
            if $\Delta\mathcal{I}(e|\mathcal{S}; \mathcal{R}_{i-1}) > \Delta\mathcal{I}_m$ then
                $\Delta\mathcal{I}_m \leftarrow \Delta\mathcal{I}(e|\mathcal{S}; \mathcal{R}_{i-1})$;
                $e_m \leftarrow e$;
            end
        end
    end
    $\mathcal{R}_i \leftarrow \mathcal{R}_{i-1} \cup \{e_m\}$;
    $\mathcal{E} \leftarrow \mathcal{E} \cup \{e_m\}$;
    $C \leftarrow C \setminus \{e_m\}$;
end
$\mathcal{R} \leftarrow \mathcal{R}_N$;
return $\mathcal{R}$.
```

also be high, especially when the size of C is very large. To further improve its computational efficiency, we use the upper bound of the incremental influence spread of the seed set $\mathcal{S}$ to avoid calculating the incremental influence spread for certain candidate edges. Let $\Delta\mathcal{I}(e|\mathcal{S}; \mathcal{R})$ represent the incremental influence spread of the seed set $\mathcal{S}$ with the edge set $\mathcal{R}$ after adding a new edge e to the graph $\mathcal{G}$. Because the restricted influence spread function is submodular, $\Delta\mathcal{I}(e|\mathcal{S}; \mathcal{R})$ is upper bounded by $\Delta\mathcal{I}(e|\mathcal{S}; \mathcal{R}')$ for the edge e, where $\mathcal{R}' \subseteq \mathcal{R}$. Therefore, we can get the following lemma.

Lemma 5.5. *If the incremental influence spread of the seed set $\mathcal{S}$ with an edge set $\mathcal{R}$ for a new added edge in the previous iterations is no more than the maximum incremental influence spread in the current iteration, then the edge can be pruned in this iteration.*

According to the above Lemma 5.5, we only need to preferentially calculate the incremental influence spread of $\mathcal{S}$ for some candidate edges, whose upper bound values are larger than the currently maximum incremental influence spread in each iteration. By making use of Lemmas 5.4 and 5.5, it is able to significantly improve the efficiency in the edge selection.

Algorithm 2 presents the pseudocode of the improved greedy algorithm. The algorithm takes a directed graph $\mathcal{G}$, a seed set $\mathcal{S}$ of size K, a candidate edge set C and the number of edges N as input, and outputs the edge set $\mathcal{R}$ with the largest increased influence spread of $\mathcal{S}$ in the graph



(a)

(v_i, v_j)	$i = 1$			$i = 2$			$i = 3$		
	$\mathcal{I}(\mathcal{S};\mathcal{R})$	$\Delta\mathcal{I}(\mathcal{S};\mathcal{R})$	$\Delta\mathcal{I}_m$	$\mathcal{I}(\mathcal{S};\mathcal{R})$	$\Delta\mathcal{I}(\mathcal{S};\mathcal{R})$	$\Delta\mathcal{I}_m$	$\mathcal{I}(\mathcal{S};\mathcal{R})$	$\Delta\mathcal{I}(\mathcal{S};\mathcal{R})$	$\Delta\mathcal{I}_m$
(v_1, v_2)	1.2	0.2	0.2	1.44	0.44	**0.2**	-	-	-
(v_2, v_3)	-	-	0.2	-	-	0.2	1.44	0.44	0
(v_1, v_3)	1.24	0.24	**0.24**	-	-	-	-	-	-
(v_1, v_4)	1.2	0.2	0.24	1.4	0.4	0.2	1.6	0.6	**0.16**
(v_3, v_2)	-	-	0.24	1.24	0.24	0.2	-	-	0.16
(v_4, v_2)	-	-	0.24	1.24	0.24	0.2	-	-	0.16

(b)

Fig. 4. The example of the improved greedy algorithm for the seed set $\mathcal{S} = \{v_1\}$.

$\mathcal{G}$. Let us define $\Delta\mathcal{I}_m$ as a dynamic influence threshold, which stores the maximum incremental influence spread of $\mathcal{S}$ in the current iteration and is updated in each iteration. The algorithm also selects all edges in an iterative fashion. More specifically, in the i iteration, for any candidate edge e in C, if its start node cannot be reachable from $\mathcal{S}$, which means that it is an unpromising candidate edge based on Lemma 5.4, the edge can be filtered out. In addition, if the edge e has been checked in the previous iterations and its incremental influence spread is not larger than $\Delta\mathcal{I}_m$ in the current iteration, then it also can be pruned based on Lemma 5.5. Otherwise, the algorithm needs to calculate the incremental influence spread of $\mathcal{S}$ with the edge set $\mathcal{R}_{i-1}$ for the edge e, and inserts the edge e and its incremental influence spread $\Delta\mathcal{I}(e|\mathcal{S};\mathcal{R}_{i-1})$ into the corresponding queue $\mathcal{M}_i$. Moreover, if the incremental influence spread $\Delta\mathcal{I}(e|\mathcal{S};\mathcal{R}_{i-1})$ is more than $\Delta\mathcal{I}_m$, then it needs to update $\Delta\mathcal{I}_m$ and the edge e_m. The algorithm continues to check the remaining edges in C similar to the above process. Therefore, it is able to select the optimal edge in this iteration, and add the edge into $\mathcal{R}_i$. When the number of edges in $\mathcal{R}$ is N, the algorithm terminates and returns the final edge set $\mathcal{R}$.

Let us take the following example to further illustrate how the improved greedy algorithm selects the edges more efficiently. In Figure 3, we assume that the set C contains $\{(v_1, v_2), (v_2, v_3), (v_1, v_3), (v_1, v_4), (v_3, v_2), (v_4, v_2)\}$ and the size of $\mathcal{R}$ is 3. Figure 4 shows an example of the improved greedy algorithm to select the edge set $\mathcal{R}$ for the seed set $\mathcal{S} = \{v_1\}$. Specifically, in the first iteration, the algorithm checks the candidate edges (v_1, v_2), (v_1, v_3), and (v_1, v_4). According to Lemma 5.4, due to the nodes v_2, v_3, and v_4 cannot be reachable from the seed set $\mathcal{S}$, it does not need to calculate the edges (v_2, v_3), (v_3, v_2), and (v_4, v_2) in this iteration actually. Therefore, it can select the first edge (v_1, v_3) that achieves the maximal increased influence spread of $\mathcal{S}$ and is added to the graph. In the second iteration, it continues to check the remaining candidate edges and can select the next edge (v_1, v_2). For the third iteration, because of Lemma 5.5, it is also not necessary to calculate the edges (v_3, v_2) and (v_4, v_2). Therefore, it can get the edge (v_1, v_4) in this iteration. Finally, the improved greedy algorithm returns the final edge set $\mathcal{R} = \{(v_1, v_3), (v_1, v_2), (v_1, v_4)\}$.

From Algorithm 2, we can see that its time complexity is also $O(N|C|\mathcal{T}(\Delta\mathcal{I}(\mathcal{S};\mathcal{R})))$. However, by Lemmas 5.4 and 5.5, the number of checked edges in each iteration is much smaller than $|C|$ in this algorithm actually. Furthermore, as a result of those optimization strategies are able to effectively filter out many unpromising candidate edges in each iteration to greatly accelerate the edge selection, the improved greedy algorithm still maintains the solution quality of the greedy algorithm, and the submodularity ratio of the restricted influence spread function in this algorithm is also 1. Therefore, the improved greedy algorithm can also solve the BIM problem with an approximation ratio of $1 - 1/e$.

Table 2. The Statistics of Four Social Networks

Networks	Email	NetHEPT	Google	Web
No. of Nodes	1005	15K	16K	109K
No. of Edges	26K	32K	171K	255K
Average Degree	50.9	4.23	21.7	4.68
Average Clustering Coefficient	0.3994	0.4089	0.0133	0.2346

6 EXPERIMENTS

We conduct extensive experiments over real-world available social networks with different scales to evaluate the performance of the proposed methods, meanwhile compare them with a few other baseline methods. We measure the performance of the proposed methods on various metrics. Additionally, we also study the affect of the important parameters on their performance.

6.1 Experimental Setup

In this section, we first introduce four real-world available social network datasets. Then, we present all evaluated algorithms. Finally, we reasonably set the parameters used in the mentioned algorithms. The code is implemented for each evaluated algorithm in C++ language using the Standard Template Library (STL), and all the experiments are run on windows machine with an Intel Core 3.30-GHz CPU and 24-GB memory.

6.1.1 Experimental Datasets. Four real-world social network datasets [38] of increasing sizes and different structural features are used in the experiments. The basic statistics of the four social networks[2] are summarized in Table 2. The first network dataset is Email network, where nodes represent users and a directed edge from u to v represents user u sends at least one email to user v. The second network dataset, NetHEPT, is an academic collaboration network, which is extracted from the "High Energy Physics-Theory" section in arXiv. Nodes in NetHEPT represent authors and edges represent the coauthorship relations. It considers that the network is directed by allowing each edge in both directions. The third network dataset is Google network, where nodes represent pages of the site google.com and directed edges represent hyperlinks between different pages. The last network dataset is Web network, where nodes represent webpages and directed edges represent hyperlinks between them.

6.1.2 Evaluated Algorithms. To evaluate the performance, we compare our proposed methods with several other heuristic strategies. All edge selection methods are presented as follows.

- Large Out-Degree (LOD) method. We select N edges whose end nodes have the largest out-degree [17] in a graph from the candidate edge set.
- Random method. We randomly select N edges from the candidate edge set, which acts as a baseline method.
- Largest Activation Probability (LAP) method. We select N edges with the largest activation probability from the candidate edge set, which also acts as a baseline method.
- Simple Greedy Algorithm (SGA). We apply the simple greedy algorithm to select the edges from the candidate edge set.
- Improved Greedy Algorithm (IGA). Instead of the simple greedy algorithm, we use the improved greedy algorithm that integrates several optimization strategies.

[2] We do not use very large social networks in the experiments. It mainly considers that the number of edges in these networks is usually very large, the increased influence spread of the seed set may very little when a limited number of new edges are added, and the figures of results of the increased influence spread may also not be distinguishable.

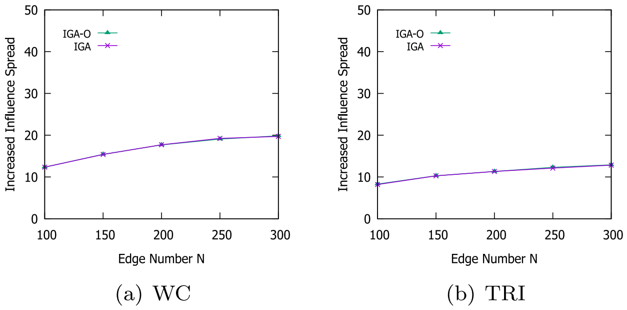


Fig. 5. The results of the increased influence spread for the influential nodes over the Email social network.

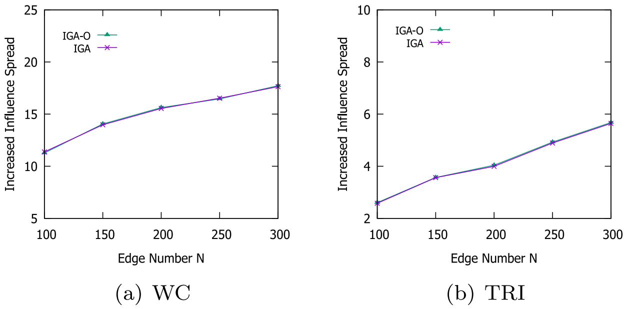
6.1.3 Parameters Setting. For most of influence maximization problems, the activation probability on each edge is calculated mainly based on the following two approaches. One approach is the weighted cascade model (WC) proposed in Reference [23]. The activation probability on the edge (u, v) is calculated as $\omega(u, v) = 1/d(v)$, where $d(v)$ is the indegree of node v. The other approach is the Trivalency model (TRI) proposed in Reference [6], which chooses globally constant randomly from the predefined set $\{0.1, 0.01, 0.001\}$, where the elements correspond to high, medium and low influences, respectively. In the experiments, we apply these two approaches to capture the activation probability on each edge.

To simulate the BIM problem, we randomly pick a subset of edges from $\mathcal{E}$ as the candidate edge set C, and remove the selected edges from the graph $\mathcal{G}$. The size of the set C is set to 2,000. Additionally, we select K seed nodes by using the efficient MIA method (i.e., the influential nodes) or the random method (i.e., the random nodes) in the refined graph. When evaluating the increased influence spread of the seed set with a new added edge set in the MIA method, the threshold parameter θ is set to 0.001 empirically to achieve the tradeoff between the calculation of the influence spread and running time.

6.2 Experimental Results and Analysis

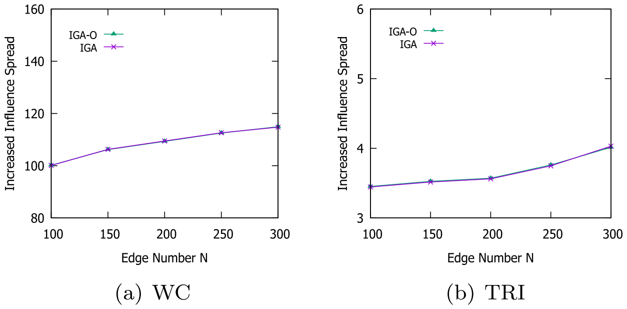
In this section, we present experimental results and analysis for all evaluated methods over the four social networks. We evaluate the performance of each method on various metrics such as the quality of edge set, running time, and so on. Furthermore, we also evaluate the affect of some important parameters in the social networks.

6.2.1 The Increased Influence Spread for Different Influence Spread Functions. We first evaluate the performance of the increased influence spread for IGA method between the original influence spread function (i.e., IGA-O) and restricted influence spread function (i.e., IGA) in the social networks. Figures 5 and 6 show the results of the increased influence spread for different kinds of seed nodes and activation probabilities over the Email social network when the K is set to 50. Figures 7 and 8 show the results over the NetHEPT social network. More specifically, we can clearly observe from Figure 5 that for the influential nodes, as the size of the selected edge set N increases, the increased influence spread for IGA method is similar to that of IGA-O method whatever the activation probability is WC model or TRI model. From Figure 6, we can observe the similar results, i.e., when the activation probability is WC model or TRI model, the increased influence spread of the random nodes for IGA method is also similar to those results for IGA-O method with the N grows. In addition, it can find from these two figures that the increased influence spread of the influential nodes and random nodes under the WC model is generally larger than the increased influence spread under the TRI model for both IGA and IGA-O methods. Specially, the results of



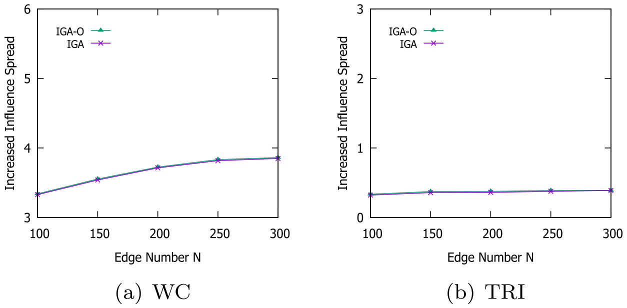
(a) WC (b) TRI

Fig. 6. The results of the increased influence spread for the random nodes over the Email social network.



(a) WC (b) TRI

Fig. 7. The results of the increased influence spread for the influential nodes over the NetHEPT social network.

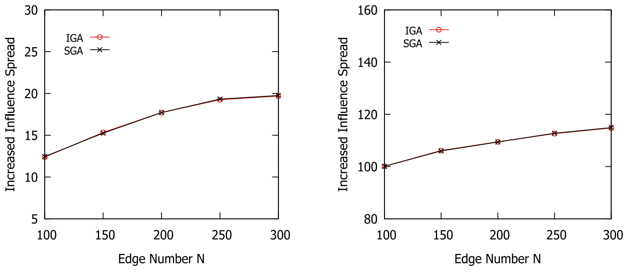


(a) WC (b) TRI

Fig. 8. The results of the increased influence spread for the random nodes over the NetHEPT social network.

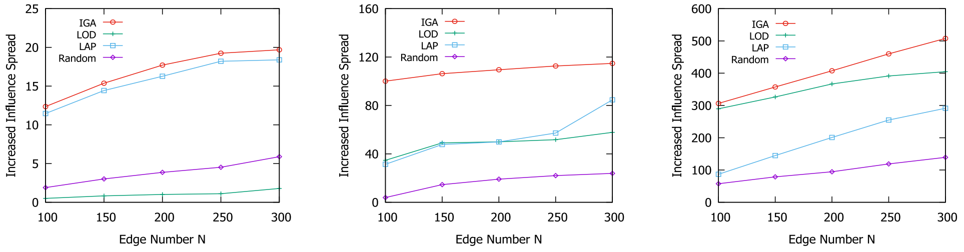
the increased influence spread over the NetHEPT social network, which is presented in Figures 7 and 8, are similar to those results in Figures 5 and 6. Therefore, it can conclude from these results in Figures 5–8 that the restricted influence spread can get a near-optimal edge set for the BIM problem.

6.2.2 Comparisons of SGA and IGA Methods on Increased Influence Spread. To investigate that IGA method does not affect the effectiveness of SGA method on the edge selection for the BIM problem, we evaluate the performance of the increased influence spread for these two methods in the social networks. Specifically, Figure 9 shows the results of the increased influence spread of the influential nodes under the WC model over the Email and NetHEPT social networks when the K is 50. From Figure 9, we can observe that IGA method achieves almost the same increased



(a) Email (b) NetHEPT

Fig. 9. The results of the increased influence spread for SGA and IGA methods over the Email and NetHEPT social networks.



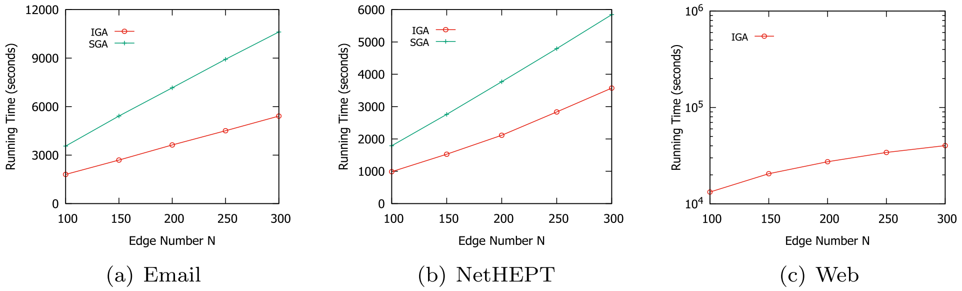
(a) Email (b) NetHEPT (c) Web

Fig. 10. The results of the increased influence spread with $K = 50$ over the social networks.

influence spread as the computationally expensive SGA method when the N gradually increases. It demonstrates that the optimization strategies in IGA method does not affect the quality of the solution generated by SGA method in practice. Therefore, when evaluating the increased influence spread of a seed set in the experiments, we do not compare SGA method.

6.2.3 Quality of Edge Set. The quality of edge set is evaluated mainly based on the increased influence spread of a seed set for a new added edge set in the social networks. It means that the larger the increased influence spread, the higher its quality. To make sure that each evaluated method is compared in a fair and accurate way, we first apply 10,000 Monte Carlo simulations with the edge set selected by each method to get an accurate estimation, then the average value is used as the final increased influence spread. Specially, the influential nodes are used for all evaluated methods under the WC model.

Figure 10 shows the results of the increased influence spread of the influential nodes over the social networks when the K is set to 50. For each evaluated method, the N increases from 100 to 300. More specifically, we can find from Figure 10 that as we expected, the increased influence spread for each evaluated method maintains a gradually increasing trend with the N grows. It keeps in line with the actual situations where a larger number of new edges to be added tends to achieve the larger increased influence spread in the social networks. Furthermore, we can also find from Figure 10 that IGA method achieves the largest increased influence spread in all evaluated methods. Accordingly, it is able to verify the effectiveness of IGA method in selecting the edge set for the BIM problem. Meanwhile, LOD, LAP and Random methods achieve the relatively lower increased influence spread with the increase of the N in the social networks. Therefore, we can conclude from these results that the proposed IGA method outperforms other compared methods and can be effectively applied to solve the BIM problem in social networks.



(a) Email (b) NetHEPT (c) Web

Fig. 11. The results of running time with $K = 50$ over the social networks.

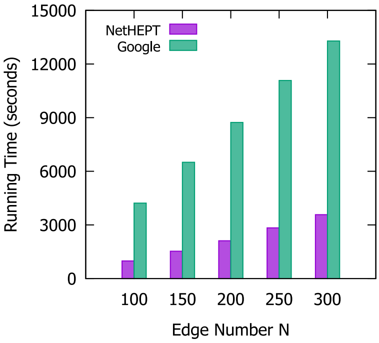
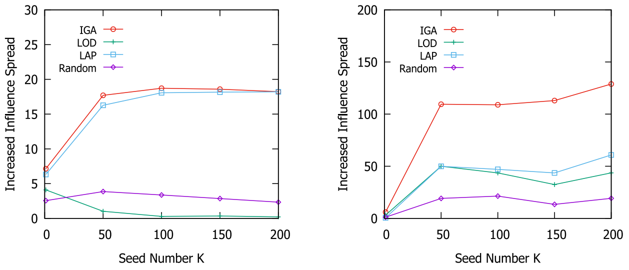


Fig. 12. The result of the running time for IGA method with $K = 50$ over the NetHEPT and Google social networks.

6.2.4 Running Time. Figure 11 shows the time taken by the evaluated methods over the social networks when the N changes. As a result of the running time of LOD, LAP, and Random methods are too trivial, which finish almost instantly for all social networks, we do not include them to make figures more distinguishable. In Figure 11, the time consumption of IGA and SGA methods are also increasing with the N grows for all social networks. Due to SGA method takes too long time to finish for the relatively large social networks, it is excluded from the Web social network. Furthermore, we can also find that IGA method runs much faster than SGA method, and is about several times more efficient than SGA method generally. Therefore, it can further verify the efficiency of the proposed IGA method for solving the BIM problem in the social networks. The main reason is the fact that IGA method makes the most of several effective optimization strategies, which are able to prune many unpromising candidate edges to narrow the space of the set C in each iteration and avoid repeatedly computing the incremental influence spread for all candidate edges exactly. Moreover, it applies an efficient heuristic method to estimate the increased influence spread of a seed set in the social networks.

6.2.5 The Efficiency for Different Kinds of Networks. To study the efficiency of the proposed methods in different kinds of networks, we evaluate the performance of the running time for IGA method over the NetHEPT and Google social networks. This is mainly because these two networks have different structural features, i.e., they have a similar number of nodes but the number of their edges vary greatly. In this situation, compared with the low-degree NetHEPT network, the high-degree Google network has a larger average degree for each node, it means that the nodes in this network are connected more closely. Specifically, Figure 12 shows the results of the running time for IGA method with different N over the NetHEPT and Google social networks when the seed



(a) Email (b) NetHEPT

Fig. 13. The results of the increased influence spread for different K over the Email and NetHEPT social networks.

nodes is the influential nodes under the WC model. From Figure 12, we can observe that the time consumption of IGA method for the Google network is much larger than the time consumption for the NetHEPT network when the N increases, which demonstrates that the efficiency of IGA method for the graphs with low-degree and high-degree features is very different. The main reason is that the pruning strategies in IGA method for these two kinds of networks are different. For the high-degree Google network, it is able to filter out less unpromising candidate edges than the low-degree NetHEPT network. As a result, IGA method needs to repeatedly check a larger number of candidate edges in the Google network and compute their incremental influence spread exactly in each iteration.

6.2.6 The Affect of Varying K on Increased Influence Spread. We further investigate the affect of the size of the seed set K on the increased influence spread for a new added edge set in the social networks. Figure 13 shows the results of the increased influence spread with varying K over the Email and NetHEPT social networks when the size of the added edge set N is fixed to 200. For each evaluated method, we select the influential nodes under the WC model, and the K takes value from 1 to 200. We can observe from Figure 13 that IGA method achieves the largest increased influence spread in all evaluated methods in the two social networks. Generally, both LOD and Random methods have the relatively lower increased influence spread than LAP method. However, the increased influence spread for some evaluated methods may decrease with the K increases in both social networks. This may be because when the number of seed nodes becomes larger, it is more likely to occur the influence blocking between different seed nodes in the influence diffusion. In addition, we can also find that the increased influence spread in the NetHEPT social network is generally larger than those results in the Email social network whatever the value of K is. The main reason may be that the average degree of nodes in the NetHEPT network is much less than the Email network, the effects of the increased influence spread of the influential nodes in the NetHEPT network may be more significant with the K grows when a number of new edges are added. Therefore, it demonstrates that the size of the seed set also has an important impact on the increased influence spread for the BIM problem in the social networks.

7 CONCLUSION AND FUTURE WORK

In this work, we address the BIM problem from a novel edge-level prospective in social networks. As a result of the BIM problem is NP-hard and the influence spread function is monotone but non-submodular, we develop a restricted form of the influence spread function, and propose a greedy algorithm for solving the problem effectively. To further improve its computational efficiency, we propose an improved greedy algorithm to greatly accelerate the edge selection. The extensive experiments over real-world available social networks of different sizes and structural features demonstrate that the proposed methods can achieve high performance on various metrics such as the influence spread, running time.

This work also inspires us a number of extensions and promising research directions for future work. First, for the BIM problem, we consider studying other goals such as finding the minimum size edge set that is added to a network such that the increased influence spread of a seed set is not less than a given threshold. Second, as the size of the network is increasing in reality, it is very essential to design more scalable heuristics for very large social networks. Last, it is deserve to further explore how to maximize the influence spread by selecting the initial adopters and the added edges at the same time under a limited budget.

REFERENCES

[1] Stefanos Antaris, Dimitrios Rafailidis, and Alexandros Nanopoulos. 2014. Link injection for boosting information spread in social networks. *Soc. Netw. Anal. Min.* 4, 1 (2014), 1–16.

[2] Ilija Bogunovic, Junyao Zhao, and Volkan Cevher. 2018. Robust maximization of nonsubmodular objectives. In *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics (AISTATS'18).*

[3] Christian Borgs, Michael Brautbar, Jennifer Chayes, and Brendan Lucier. 2014. Maximizing social influence in nearly optimal time. In *Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms*, 946–957.

[4] Erik Cambria, Marco Grassi, Amir Hussain, and Catherine Havasi. 2012. Sentic computing for social media marketing. *Multimedia Tools Appl.* 59, 2 (2012), 557–577.

[5] Vineet Chaoji, Sayan Ranu, Rajeev Rastogi, and Rushi Bhatt. 2012. Recommendations to boost content spread in social networks. In *Proceedings of the 21st ACM International Conference on World Wide Web*, 529–539.

[6] Wei Chen, Chi Wang, and Yajun Wang. 2010. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1029–1038.

[7] Wei Chen, Yifei Yuan, and Li Zhang. 2011. Scalable influence maximization in social networks under the linear threshold model. In *Proceedings of the 10th IEEE International Conference on Data Mining*, 88–97.

[8] Yi-Cheng Chen, Wen-Yuan Zhu, Wen-Chih Peng, Wang-Chien Lee, and Suh-Yin Lee. 2014. CIM: Community-based influence maximization in social networks. *ACM Trans. Intell. Syst. Technol.* 5, 2 (2014), 1–31.

[9] Judith A. Chevalier and Dina Mayzlin. 2006. The effect of word-of-mouth on sales: Online book reviews. *J. Market. Res.* 43, 3 (2006), 345–354.

[10] Pierluigi Crescenzi, Gianlorenzo D'angelo, Lorenzo Severini, and Yllka Velaj. 2016. Greedily improving our own closeness centrality in a network. *ACM Trans. Knowl. Discov. Data* 11, 1, Article 9 (2016), 32 pages.

[11] Gianlorenzo D'Angelo, Lorenzo Severini, and Yllka Velaj. 2019. Recommending links through influence maximization. *Theoretical Computer Science* 764 (2019), 30–41.

[12] Abhimanyu Das and David Kempe. 2011. Submodular meets spectral: Greedy algorithms for subset selection, sparse approximation and dictionary selection. In *Proceedings of International Conference on Machine Learning*, 1057–1064.

[13] Erik D. Demaine and Morteza Zadimoghaddam. 2010. Minimizing the diameter of a network using shortcut edges. In *Proceedings of theScandinavian Conference on Algorithm Theory*, 420–431.

[14] Pedro Domingos and Matthew Richardson. 2001. Mining the network value of customers. In *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 57–66.

[15] Fabrizio Frati, Serge Gaspers, Joachim Gudmundsson, and Luke Mathieson. 2015. Augmenting graphs to minimize the diameter. *Algorithmica* 72, 4 (2015), 995–1010.

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

[17] Chao Gao, Jiming Liu, and Ning Zhong. 2011. Network immunization and virus propagation in email networks: experimental evaluation and analysis. *Knowl. Inf. Syst.* 27, 2 (2011), 253–279.

[18] Arpita Ghosh and Stephen Boyd. 2006. Growing well-connected graphs. In *Proceedings of the IEEE Conference on Decision and Control*, 6605–6611.

[19] Jacob Goldenberg, Barak Libai, and Eitan Muller. 2001. Talk of the network: A complex systems look at the underlying process of word-of-mouth. *Market. Lett.* 12, 3 (2001), 211–223.

[20] Amit Goyal, Wei Lu, and Laks V. S. Lakshmanan. 2011. Celf++: Optimizing the greedy algorithm for influence maximization in social networks. In *Proceedings of the ACM International Conference Companion on World Wide Web*, 47–48.

[21] Kyomin Jung, Wooram Heo, and Wei Chen. 2013. IRIE: Scalable and robust influence maximization in social networks. In *Proceedings of the IEEE International Conference on Data Mining*, 918–923.

[22] Richard Karp. 2010. Reducibility among combinatorial problems. *J. Symbol. Logic* 40, 4 (2010), 618–619.

[23] David Kempe, Jon Kleinberg, and Éva Tardos. 2003. Maximizing the spread of influence through a social network. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 137–146.

[24] Masahiro Kimura, Kazumi Saito, and Hiroshi Motoda. 2008. Minimizing the spread of contamination by blocking links in a network. In *Proceedings of the 23rd National Conference on Artificial Intelligence*, 1175–1180.

[25] Masahiro Kimura, Kazumi Saito, and Hiroshi Motoda. 2008. Solving the contamination minimization problem on networks for the linear threshold model. *Malay. J. Med. Sci.* 12, 2 (2008), 50–55.

[26] Masahiro Kimura, Kazumi Saito, and Hiroshi Motoda. 2009. Blocking links to minimize contamination spread in a social network. *ACM Trans. Knowl. Discov. Data* 3, 2 Article 9 (2009), 23.

[27] Chris J. Kuhlman, Gaurav Tuli, Samarth Swarup, Madhav V. Marathe, and S. S. Ravi. 2013. Blocking simple and complex contagion by edge removal. In *Proceedings of the IEEE International Conference on Data Mining*, 399–408.

[28] Jure Leskovec, Andreas Krause, Carlos Guestrin, Christos Faloutsos, Natalie Glance, and Natalie Glance. 2007. Cost-effective outbreak detection in networks. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 420–429.

[29] Yanhua Li, Wei Chen, Yajun Wang, and Zhi-Li Zhang. 2013. Influence diffusion dynamics and influence maximization in social networks with friend and foe relationships. *Proceedings of the ACM International Conference on Web Search and Data Mining*, 657–666.

[30] Elchanan Mossel and Sebastien Roch. 2007. On the submodularity of influence in social networks. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing*. 128–134.

[31] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. 1978. An analysis of the approximations for maximizing submodular set functions. *Math. Program.* 14, 1 (1978), 265–294.

[32] M. E. J. Newman. 2003. The structure and function of complex networks. *SIAM Rev.* 45, 2 (2003), 167–256.

[33] Hung T. Nguyen, My T. Thai, and Thang N. Dinh. 2016. Stop-and-stare: Optimal sampling algorithms for viral marketing in billion-scale networks. In *Proceedings of the ACM International Conference on Management of Data*, 695–710.

[34] Manos Papagelis. 2015. Refining social graph connectivity via shortcut edge addition. *ACM Trans. Knowl. Discov. Data* 10, 2, Article 12 (2015), 35 pages.

[35] Guido Proietti. 2012. Improved approximability and non-approximability results for graph diameter decreasing problems. *Theor. Comput. Sci.* 417, 1 (2012), 12–22.

[36] Khadije Rahimkhani, Abolfazl Aleahmad, Maseud Rahgozar, and Ali Moeini. 2015. A fast algorithm for finding most influential people based on the linear threshold model. *Exp. Syst. Appl.* 42, 3 (2015), 1353–1361.

[37] Matthew Richardson and Pedro Domingos. 2002. Mining knowledge-sharing sites for viral marketing. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 61–70.

[38] SNAP Datasets. 2014. Retrieved from http://snap.stanford.edu/data/.

[39] Youze Tang, Yanchen Shi, and Xiaokui Xiao. 2015. Influence maximization in near-linear time: A martingale approach. In *Proceedings of ACM SIGMOD International Conference on Management of Data*, 1539–1554.

[40] Youze Tang, Xiaokui Xiao, and Yanchen Shi. 2014. Influence maximization: Near-optimal time complexity meets practical efficiency. In *Proceedings of ACM SIGMOD International Conference on Management of Data*, 75–86.

[41] Yu Wang, Gao Cong, Guojie Song, and Kunqing Xie. 2010. Community-based greedy algorithm for mining top-k influential nodes in mobile social networks. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1039–1048.

[42] Chuan Zhou, Peng Zhang, Wenyu Zang, and Li Guo. 2014. Maximizing the long-term integral influence in social networks under the voter model. In *Proceedings of the ACM International Conference on World Wide Web*, 423–424.

Received March 2019; revised July 2019; accepted September 2019