Computer Communications 212 (2023) 182–197

Contents lists available at ScienceDirect

Computer Communications

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

Minimizing cost for influencing target groups in social network: A model and algorithmic approach

Phuong N.H. Pham [a], Canh V. Pham [b,*], Hieu V. Duong [c], Václav Snášel [d], Nguyen Trung Thanh [b]

[a] Faculty of Information Technology, Ho Chi Minh City University of Industry and Trade, Ho Chi Minh, Viet Nam
[b] ORLab, Phenikaa University, Hanoi, 12116, Viet Nam
[c] People's Security Accedemy, Hanoi, Viet Nam
[d] Faculty of Electrical Engineering and Computer Science, VSB-Technical University of Ostrava, Ostrava, Czech Republic

ARTICLE INFO

Keywords:
Online social networks
Social influence
Group influence
Approximation algorithms

ABSTRACT

Stimulated by practical applications arising from economics, viral marketing, and elections, this paper studies the problem of Groups Influence with Minimum cost (GIM), which aims to find a seed set with the smallest cost that can influence all target groups in a social network, where each user is assigned a cost and a score and a group of users is influenced if the total score of influenced users in the group is at least a certain threshold. As the group influence function, defined as the number of influenced groups or users, is neither submodular nor supermodular, theoretical bounds on the quality of solutions returned by the well-known greedy approach may not be guaranteed.

In this work, two efficient algorithms with theoretical guarantees for tackling the GIM problem, named Groups Influence Approximation (GIA) and Exact Groups Influence (EGI), are proposed. GIA is a bi-criteria polynomial-time approximation algorithm and EGI is an (almost) exact algorithm; both can return good approximate solutions with high probability. The novelty of our approach lies in two aspects. Firstly, a novel group reachable reverse sample concept is proposed to estimate the group influence function within an error bound. Secondly, a framework algorithmic is designed to find serial candidate solutions with checking theoretical guarantees at the same time. Besides theoretical results, extensive experiments conducted on real social networks show our algorithms' performance. In particular, both EGI and GIA provide the solution quality several times better, while GIA is up to 800 times faster than the state-of-the-art algorithms.

1. Introduction

Information propagation and Influence Maximization (IM) in Online Social Networks (OSNs) have been hot research topics recently due to their wide range of applications in the commercial. Nowadays, organizations and companies have used social media platforms as practical tools to promote products, spread renovations and ideas, persuade voters, etc.

In Kempe et al. [1], a paper published almost twenty years ago, introduced the problem of IM, which aimed at finding a set of some key users (called *seed set*) in an online social network to start a process that could possibly influence the largest number of users. Since then, IM problem has demonstrated its important role in various domains, not only in product promotion and social influence [2,3], but also in other applications such as social network monitoring [4–6], epidemic presentation [7–10] and recommendation system [11].

In some realistic scenarios, the decision and behavior of a user depend on one's team, and a group of essential persons may affect many individuals making their important decisions. Therefore, exerting an impact on groups or communities of users has more benefits than that on each individual and deserves special consideration.

A typical example is the US Presidential election, where any candidate who gets an absolute majority of electoral votes (not the popular votes) will be selected as the winner of the presidency. In reality, the candidate often focuses on swing States to get the last win. Each State is allocated a fixed number of electoral votes that a candidate can own if one wins the most significant popular votes in the State. To do this, the candidate's election campaigns might leverage social networks in order to persuade the most voters in the State. As the budget of the campaign is limited, the candidate would not be able to convince all of the voters in the State, and one should target the most influenced voters (some key persons) in the State. Another example is about combating misinformation or rumors in social networks. To protect some groups of users in a social network against the attack of misinformation or

* Corresponding author.
E-mail address: canh.phamvan@phenikaa-uni.du.vn (C.V. Pham).

https://doi.org/10.1016/j.comcom.2023.09.022
Received 21 July 2022; Received in revised form 1 September 2023; Accepted 19 September 2023
Available online 22 September 2023
0140-3664/© 2023 Elsevier B.V. All rights reserved.

rumor, one can select some seed users to initiate the good or official information to influence groups before bad effects influence them.

Motivated by the aforementioned phenomena, recent studies have been focused on the problem of group influence maximization, which asks to find some critical users that influence the largest number of groups or communities of users in an online social network (see, e.g., [12–15]). It is obvious that the mentioned problem is a generalized version of IM, and thus it is computationally hard. In the two above examples and some other contexts, some target groups in a social network are required to be influenced, but the selection of seed sets for influencing the group with limited cost or size constraints in the previous studies seems inefficient because it may not influence all groups or select too many unnecessary seed nodes. Consequently, in this scenario, the campaign that ensures all groups are influenced with minimal cost is more efficient than the previous strategies. Therefore, one can consider a dual problem of this problem by asking for the minimum number of users to influence a given set of groups. From this point of view, this paper investigates a slightly general problem named GIM, which asks to find a set of key users with a minimum total cost to influence all the given target groups in an online social network.

Different from existing works, in this paper, each user's role in a group is first considered by a score, and each group has a threshold representing how difficult it influences that group. Besides, in this context, we also constrain a general rule for influencing a group: a group is influenced if the total score of affected members reaches at least its threshold. It can be easily seen that GIM is a dual version of influence group maximization. Thus, it is **NP**-hard to solve, not only by the combinatorial structure of the problem but also by the #**P**-hardness of the calculation of the number of influence groups (denoted by $\sigma(\cdot)$ function). Another challenge for solving the considered problem is that $\sigma(\cdot)$ is neither a submodular nor supermodular, implying that GIM does not admit the traditional greedy methods with any theoretical bound.

Our Contributions. Assuming that $G = (V, E)$ is modeled as a social network and $\mathcal{C} = \{C_1, C_2, \ldots, C_K\}$ is a set of target groups, $C_i \subseteq V, \forall i \in [K]$. Each group $C_i \in \mathcal{C}$ has a certain threshold $t_i > 0$, and each node $u \in V$ is assigned a positive cost $c(u)$ and a positive score $b(u)$. In this work, we address the aforementioned challenges for addressing the GIM problem. A preliminary version of this work appears in the proceedings of the 10th International Conference on Computational Data and Social Networks [16]). This work extends and revises the conference version by providing all the proofs, more algorithm, experiment evaluation and discussions. Our contributions can be summarized as follows.

1. We first model the process of influencing groups and formulate the group influence function in a more rational and general way than existing studies. Based on that, we formulate the GIM problem and show that the influence function $\sigma(\cdot)$ is neither submodular nor supermodular, and cannot apply the greedy algorithms with any theoretical guarantee.
2. We show that calculating $\sigma(\cdot)$ is #P-hard. To estimate $\sigma(\cdot)$ within an error bound, we propose a novel concept of sampling technique, named *Group Reachable Reverse* (GRR), that plays an essential role in our proposed algorithmic framework.
3. We first devise a bi-criteria approximation algorithm GIA for the proposed algorithms. In particular, GIA is a $(O(\ln(K)+\ln\ln(n)), 1-\epsilon)$-bicriteria approximation with a high probability (w.h.p), that is, it provides in polynomial time a solution S such that $c(S) \leq O(\ln(K) + \ln\ln(n))\mathsf{OPT}$ and $\sigma(S) \geq (1-\epsilon)K$ w.h.p, where OPT is the total cost of an optimal solution and $\epsilon > 0$ is an accuracy parameter. The key of our algorithm lines in a novel algorithmic framework to operate in the multiple iterations in which each generating candidate solution with checking solution quality.
4. We further propose an (almost) exact algorithm EGI, which returns $c(S) \leq \mathsf{OPT}$, and $\sigma(S) \geq (1-\epsilon)K$ w.h.p via reusing the GIA's framework and formulating an integer programming model.
5. This work illustrates some extensive experiments in real social networks to investigate the proposed algorithms' performance. The results show that our algorithms provide a significantly better solution than state-of-the-art algorithms in both solution quality and computational time. Specifically, both EGI and GIA provide seed sets that have not only total costs several times smaller but also larger values of the group influence function than existing algorithms. Besides, our EGI provides the best solution quality. Furthermore, our GIA algorithm is up to 800 times faster than state-of-the-art algorithms and can run in large networks within a few seconds.

Organization. The rest of the paper is organized as follows. Section 2 gives a literature review for the studied problem, and Section 3 presents the information diffusion model and formally introduces problem formulation. Section 4 is devoted to presenting our novel sampling technique and Section 5 presents our proposed algorithms. The experiments and results are presented in Section 6. Finally, Section 7 concludes this work and discusses future studies.

2. Related works

This section will review some previous significant works related to our studied problem GIM, including classical Influence Maximization, Groups Influence Optimization, and Non-submodular Optimization.

Influence Maximization. Inspired by the discovery of the impact among users in an OSN for the purpose of viral marketing, Kempe et al. [1] first introduced the IM problem as a combinatorial optimization problem under two information diffusion models: Independent Cascade (IC) and Linear Threshold (LT). The challenges existed in two points: (1) IM was **NP**-hard and it could not be approximated within a ratio of $1 - 1/e + \epsilon$ in polynomial time for any $\epsilon > 0$; (2) computing the influence spread (objective function) of a seed node was #**P**-hard [17,18]. The Kempe et al.'s work has inspired vast amount of later studies on investigating efficient algorithms for IM [2,17–23] as well as its variants such as influence threshold problem [24,25], location/distance-aware influence maximization [26,27], cost-aware influence maximization [2], group-aware influence maximization [12], etc.

Since IM is NP-hard, two common methods are used to solve this problem in a reasonable duration: heuristic and approximation with theoretical bounds. By utilizing the monotonicity and submodularity of the influence spread function, the authors in [1] first proposed a classical greedy algorithm that returned an approximation ratio of $1 - 1/e$. However, due to the expensive cost to compute the influence function [28], the greedy algorithm took a long time, even for a small network. Later on, Leskovec et al. [19] devised a cost-effective lazy-forward (CELF) algorithm, which was approximately 700 times faster than the greedy algorithm. Several fast heuristic algorithms, which converted the graph G to a directed acyclic graph in order to obtain a linear-time complexity of the influence spread computation, were proposed for the medium and large networks [17,18,29], without any theoretical bounds. In 2014, Borgs et al. [30] introduced a sampling method called Reverse Influence Sampling (RIS), which paved the way for the development of a $(1 - 1/e - \epsilon)$-approximation algorithm within near-linear time complexity. Recently, several works have been proposed for reducing both sampling complexity and running time by modifying the RIS model [20–23]. According to the heuristic approach, several fast heuristic algorithms, which converted the original graph to a directed acyclic graph, were proposed for finding the solution on large-scale networks. Although these algorithms significantly reduced the running time on both theoretical and practical sides, they did not provide any theoretical guarantees for obtained solutions [17,18]. The authors in [31] proposed the SIMPATH, an efficient algorithm based on estimating the influence function via a simple path over the social network. Their algorithm provided a competitive solution quality but ran

faster than algorithms in [17,18]. Several algorithms used for the divide and conquer strategy were proposed by [32,33]. They first divided the network into communities and then found and combined the seed sets over these communities. Although these algorithms significantly improved with respect to running time compared with the previous heuristic one, the quality of those solutions was not better than that of the traditional greedy algorithm. Recently, several authors have proposed meta-heuristic techniques such as swarm optimization [34], simulated annealing [35], and genetic algorithm [36]. A main drawback of these algorithms was that they may require many influence function calls, rendering them not applicable to large networks.

In other directions, several works have expanded IM for the other contexts of viral marketing and devised efficient approximation algorithms based on extending greedy method and sampling algorithms in [37]. Nguyen et al. [38] considered IM under the budget constraint (called BIM) and proposed several fast heuristic algorithms. IM with the topic query was also studied in [3,39,40]. In this problem, the information diffusion model with many topics has been introduced. The authors in [2] studied a generalization problem of both IM and BIM named Cost-aware Targeted Viral Marketing (CTVM) in which each node u had an arbitrary cost $c(u)$ and a benefit $b(u)$. The goal of CTVM was to select a set of nodes within a given budget so that the total benefit of influenced nodes was maximized. In the same paper, the authors proposed an approximation algorithm with a ratio of $(1-1/\sqrt{e}-\epsilon)$. The advance of geo-position allows OSNs to integrate a user's location, which can be leveraged in product promotion. Authors in [41] investigated the location-aware influence maximization problem, which selected some nodes that could influence the largest nodes in a given region; authors in [26] considered the role of distance among users and the promoted location in a seed selection. In addition, several other variations of IM such as competitive-aware [28,42], time-aware [29,43] have been introduced and studied. The common approach of these algorithms is based on extending the sampling model in [37] to design approximation algorithms with the ratio of $1-1/e$.

Groups Influence Optimization. The maximizing number of influenced groups (or communities) problem is one of IM's variations; thus, it is also **NP**-hard. Due to its wide range of applications, such as viral marketing, election campaigns, etc., it has gained much attention recently. In the setting of that problem, each user on social media usually belongs to a particular group, and their group influences the user's behavior. Therefore, influencing a group of users gains more benefits than individuals. Many algorithms are proposed for these problems based on approximation algorithmic approaches, such as programming mathematical and approximation algorithms via exploiting group influence function.

Nguyen et al. [12] studied the problem of Influence Maximization at the Community level (IMC), which found k users such that could influence the largest number of groups. In the seminal works, they proposed several efficient algorithms with theoretical bounds proposed under the IC model. However, the approximation ratios in their algorithms were quite small or depended on the input data. This problem is fundamentally different from our studied problem (it considers the constraint that the size seed set is at most k). Therefore, their algorithms cannot be applied directly to our studied GIM problem. We have tried to adapt their algorithms with some reasonable modifications (in combination with the binary search method) to find a solution to the GIM problem (see details in Section 6). However, their algorithms did not provide any approximation guarantees for the GIM problem.

The authors in [14,44] investigated the problem of Group Influence Maximization, which also asked to find k users that could influence the largest number of groups. They devised a sandwich approximation algorithm that had an approximation ratio depending on the value of the group function with respect to some seed sets. They considered the problem in a simpler model than [12], i.e., each group became active if β percent of nodes in this group were influenced. Therefore, their algorithm cannot be generally applied to IMC and GIM.

Table 1
Table of notations.

Notional	Description
n, m	The number of vertices and the number of edges in G, respectively
C	$C = \{C_1, C_2, \ldots, C_K\}$ is the set of target groups
$N_{out}(v)$, $N_{in}(v)$	The sets of outgoing and incoming neighbor nodes of v
C	$C = \bigcup_{i=1}^{K} C_i$
S	A seed set of our algorithms
S^*	An optimal solution of GIM problem
$C(u)$	The group that contains node u
OPT	$\mathsf{OPT} = c(S^*)$
S^0	An optimal solution of SIM problem
opt	$\mathsf{opt} = c(S^0)$
k_{max}	$k_{max} = \arg\max_{k=1\ldots n} \binom{k}{n}$
$N(\epsilon, \delta)$	$(2+\frac{2}{3}\epsilon)\frac{n}{\epsilon^2}\ln\frac{1}{\delta}$
N_{max}	$N_{max} = N(\epsilon, \delta/2\binom{K}{k_{max}})$
b_{min}, b_{max}	$b_{min} = \min_{v\in V} b(v), b_{max} = \max_{v\in V} b(v)$

More recently, Tsang et al. [13] have considered IM under several fairness constraints on groups. In seminal work, they proposed an approximation algorithmic framework with constant approximation ratios that utilized monotonic and submodular multi-objective function techniques. However, it may not work well in case the objective function is non-submodular. These problems were then solved via mixed integer programming formulation [15]. However, this approach only was applied to a set of sample graphs with fixed size (the size of sample sets was typically 500 in that paper) due to its high complexity.

In general, these studies focused on the problem of maximizing the influence of groups with a limited budget setting, which was different from the context of minimizing the costs to influence groups in the GIM problem. Therefore, the existing algorithms cannot be directly adapted to the proposed problem. In fact, the most related algorithms in [12] were applied to our work with some minor modifications.

Non-submodular optimization. This work considers the non-submodular utility (influence group) function, so this part will provide a brief review of this topic. There are many **NP**-hard combinatorial optimization problems arising from information diffusion applications that consider maximizing non-submodular function problems [45–48]. One common algorithmic approach for this type is to use the sandwich approximation algorithm, which takes advantage of the lower and upper bounds of the submodular function to give a data-dependent approximate ratio [48] data-dependent. However, this ratio is not tight and might become very small. On the other hand, several works focused on devising the approximation algorithms for non-submodular maximization by exploiting a submodularity ratio [49], a generalized curvature [50], or a diminishing-return ratio [51,52]. However, most of those works focus on non-submodular maximization problems with some constraints, which cannot be applied to the problem in this paper. For the problem, a lower bound function of a non-submodular function is designed. It is useful to exploit the group influence's properties to analyze the approximation ratios of the proposed algorithms.

3. Propagation model and problem definition

This section presents the social network model and a well-known Independent Cascade (IC) diffusion model [1]. Then the studied problem and some of its properties are provided. The frequently used notations are summarized in Table 1.

3.1. Independent cascade model

In this model, an OSN is represented by a directed graph $G = (V, E)$ where V is the set of nodes/vertices and E is the set of edges with $|V| = n$ and $|E| = m$. Let $N_{out}(v)(N_{in}(v))$ be the set of out-neighbors (in-neighbors) of node v, respectively. Given a seed set S, the process of influence propagation happens in the network in discrete

steps, and more nodes can be influenced. Independent Cascade (IC) and Linear Threshold (LT) [1] are two of the most popular models in OSNs [2,20,28,37,47]. This work only focuses on the IC model, but the model process and algorithms can be adapted for the LT model as well.

Under the IC model, each edge $e = (u, v) \in E$ has a propagation probability $p(e) \in [0, 1]$ representing the information transmission from a node u to another node v. The diffusion process happens from a seed set from S as follows.

- At the first round $t = 1$, all nodes in S are *active* and other nodes in V are *inactive*.
- At step $t > 1$, each node u activated at step $t - 1$ has a single chance to activate each currently inactive node $v \in N_{out}(u)$ with a successful probability $p(e)$.
- If a node is activated, it remains active till the end of the diffusion process. The propagation process terminates at step t if there is no new node activated in this step.

The IC model is a stochastic information propagation model. To estimate the number of influenced nodes efficiently, Kempe et al. [1] showed that the IC model is equivalent to a *live-edge* model defined as follows. From the graph $G = (V, E)$, a random sample graph g is generated from G by selecting an edge $e \in E$ with probability $p(e)$ and non selecting e with probability $1 - p(e)$. We refer to g as a sample of G and write $g \sim G$. The probability of generation a sample graph g from G is calculated by:

$$\Pr[g \sim G] = \prod_{e \in E(g)} p(e) \prod_{e \notin E(g)} (1 - p(e)) \tag{1}$$

where $E(g)$ is the set of edges in the graph g. The influence spread from a set node S to a node u is:

$$\mathbb{I}(S, u) = \sum_{g \sim G} \Pr[g \sim G] \cdot R_g(S, u) \tag{2}$$

where $R_g(S, u) = 1$ if u is reachable from S in g and $R_g(S, u) = 0$ otherwise. The influence spread of S in network G (number of influenced nodes) is:

$$\mathbb{I}(S) = \sum_{u \in V} \mathbb{I}(S, u). \tag{3}$$

3.2. Groups influence process

Given a social network $G = (V, E)$ under the IC model and a collection of $K > 0$ disjoint groups $\mathcal{C} = \{C_1, C_2, \ldots, C_K\}$ (called target groups), where $C_i \subseteq V, C_i \cap C_j = \emptyset$, for every pair (i, j) with $i \neq j$. Denote $C(u)$ the group that contains node u.

To model the process of group influence in general, the new proposed model extends the model in [12] by considering the following aspects:

- Each node u in the group C_i is assigned a positive integral *score* $b(u) \in [b_{min}, b_{max}]$ where b_{min}, b_{max} are constants. This is based on the fact that each user has a different role in their group. The node score $b(u) > 0$ measures the role of a node u in its group $C(u)$.
- Each node u has a *cost* $c(u) > 0$, which measures the cost or the price of a node u one has to pay to start influencing this node at the beginning of an influence process.
- Each group C_i is assigned an integer *threshold* $t_i > 0$, which reflects the minimum total score that the propagation must reach if it wants to influence a group C_i. The group C_i is said to be influenced if the total score of influenced nodes in C_i is at least t_i.

We define a *cost function* $c : 2^V \to \mathbb{R}_+$ with an additive property, i.e. $c(S) = \sum_{u \in S} c(u)$ is the total cost of S. A *group influence function* $\sigma : 2^V \to \mathbb{R}_+$ is defined as follows: $\sigma(S)$ is (expected) the number of groups in $\mathcal{C}$ influenced by the seed set S when the diffusion process ends, that is,

$$\sigma(S) = \sum_{g \sim G} \Pr[g \sim G] \sum_{C_i \in \mathcal{C}} I_g(S, C_i) \tag{4}$$

where $I_g(S, C_i)$ is an indicator variable that measures the influence of S to the group C_i in sample graph g, i.e,

$$I_g(S, C_i) = \begin{cases} 1, & \text{if } \sum_{u \in C_i} R_g(S, u) b(u) \geq t_i \\ 0, & \text{Otherwise.} \end{cases} \tag{5}$$

In the special case where each group C_i has only one node, the above group influence function $\sigma(\cdot)$ becomes the influence spread function $\mathbb{I}(\cdot)$ of the IM problem. As a consequence, computing $\sigma(\cdot)$ is #**P**-hard. On the other hand, one can easily verify that the function $\sigma(\cdot)$ is neither submodular nor supermodular. The function $\sigma(\cdot)$ is submodular if for every pair of subsets $A, B \subseteq V$ it holds that $\sigma(A) + \sigma(B) \geq \sigma(A \cup B) + \sigma(A \cap B)$. If the inequality holds in the reversed direction, $\sigma(\cdot)$ is a supermodular function. For completeness, a counter-example is provided in Example 1 below.

Example 1. Given a directed graph $G = (V, E)$ under IC model with $V = \{a, b, c\}$ and $E = \{(a, c), (b, c)\}$ and all edges have the same transmission probability 1. Consider a group $C = \{a, b\}$ with the threshold $t_C = 2$, we have $\sigma(\{a\}) - \sigma(\emptyset) = 0 < \sigma(\{a, b\}) - \sigma(\{a\}) = 1$, which means $\sigma(\{a\})$ is non-submodular. Also, we also have $\sigma(\{a, b, c\}) - \sigma(\{a, b\}) = 0 < \sigma(\{a, b\}) - \sigma(\{a\}) = 1$, which means $\sigma(\cdot)$ is non-supermodular.

3.3. Problem definition

This part formally defines the problem *Groups Influence with minimal Cost*, which will be studied in this paper.

Definition 1 (GIM *Problem*)**.** An instance of GIM is given by $(G, \mathcal{C})$, where $G = (V, E)$ is a social network under IC model, and $\mathcal{C}$ is a collection of K disjoint target groups $\{C_1, C_2, \ldots, C_K\}$, $C_i \cap C_j = \emptyset$. The objective is to find a seed set $S \subseteq V$ with minimum total cost that influences all the groups in $\mathcal{C}$, i.e, the problem asks to find

$$S = \arg \min_{S' \subseteq V, \sigma(S) = K} c(S').$$

The inapproximability of GIM problem states in Theorem 1, is easily obtained by reducing from the classical Set Cover problem [53].

Theorem 1. GIM *has no polynomial-time algorithm attaining an approximation ratio of* $(1 - \epsilon) \ln n$ *for any* $\epsilon > 0$*, unless* **NP** $\subset$ **DTIME**$(n^{O(\log \log n)})$.

We call an algorithm a (γ, σ)-**bicriteria approximation** for GIM problem if it returns a solution S such that $\sigma(S) \geq \gamma \cdot K$ and $c(S) \leq \sigma \cdot \mathrm{OPT}$, for $\gamma, \sigma > 0$.

4. An estimator of group influence function

This section first introduces the concept of *Group Reverse Reachable (GRR)* sample, based on the existing Reverse Reachable (RR) set [37] and Reverse Influence Community (RIC) [12] samples were captured to visualize the process of influencing a group more effectively than existing sample techniques. It then introduced the method to efficiently estimate the influence group function $\sigma(\cdot)$ on both theoretical and practical perspectives.

Definition 2 (*GRR Sample*)**.** Given a tuple $(G, \mathcal{C})$ as an instance of GIM problem. We generate a GRR sample via the following steps:

1. Randomly choose a group C_i with the probability $\frac{1}{K}$ (call C_i a *source group*).
2. Generate a sample graph g by live-edge model under IC model in [1].

3. For each vertex $u \in C_i$, return a set of nodes $R_g(u)$ which is reachable from u in the sample graph g (u is a source node).
4. Return a set $R_g = \{R_g(u) | \forall u \in C_i\}$ as a GRR sample, we denote $C(R_g)$ as the source group of R_g and $t(R_g)$ as the threshold of $C(R_g)$.

The concept of the GRR sample is an extended version of the RR set [1] by generating multiple RR sets with all source nodes in a source group. In addition, the GRR sample is also an extension of an RIC sample in [12]. The RIC sample is used to estimate the group influence with the score of each node equal to 1 in our model. The critical difference between ours and RIC lines in determining whether or not a group is influenced via the total score of influenced nodes, which was not well-defined in the RIC, even when the score is equal to 1. Storing the RR set for each node in the GRR sample is useful to exploit some helpful properties for analyzing theoretical bounds.

Given a set $S \subseteq V$ and a GRR sample R_g, and for $R_g(u) \in R_g$, if $R_g(u) \cap S \neq \emptyset$, it is said that S *covers* node u, and is defined by:

$$\mathsf{ScoreCover}(S, R_g(u)) = b(u) \cdot \min\{|S \cap R_g(u)|, 1\} \tag{6}$$

as the *score* of source node u *covered* by S in R_g. We also denote the following random variable:

$$X_g(S) = \begin{cases} 1, & \text{if } \sum_{R_g(u) \in R_g} \mathsf{ScoreCover}(S, R_g) \geq t_i \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

The variable $X_g(S)$ indicates whether the total score of nodes covered by S is greater than the threshold t_i or not. When $X_g(S) = 1$, C_i is influenced by S in the sample graph g. It is also said that a sample R_g is influenced by S. The probability of generating a sample R_g is:

$$\Pr[R_g] = \frac{1}{K} \sum_{g \sim G : reach(u,g) = R_g(u), \forall u \in C(R_g)} \Pr[g \sim G] \tag{8}$$

where $reach(u, g)$ is the set of nodes that can reach to u in g. We now show that the value of $\sigma(S)$ can be estimated by the expectation of $X_g(S)$, which is a key property of the GRR sample to keep the new algorithms with theoretical bounds.

Lemma 1. *For any set $S \subseteq V$, we have $\sigma(S, C) = K \cdot \mathbb{E}[X_g(S)]$ where the expectation is taken over the randomness of g.*

To generate a GRR sample in practice, we now introduce the Algorithm 1. Firstly, a source group C_i in $\mathcal{C}$ is randomly selected (line 1). It then generates a RR set $R_g(u)$ for each source node u in C_i by using the RIS model [20–23].

In particular, for a source node u, the algorithm uses a queue Q that consists of nodes reachable from u on the sample graph g. It updates the queue Q (line 5) and then randomly visits and adds its incoming neighbor v into Q and $R_g(u)$ with the probability $p(u, v)$. If an incoming neighbor is visited, it is added into $R_g(u)$. This process repeats until Q becomes empty.

From Lemma 1, we have an estimation of group influence function over a collection of GRR sets $\mathcal{R}$:

$$\hat{\sigma}(S) = \frac{K}{|\mathcal{R}|} \cdot \sum_{R_g \in \mathcal{R}} X_g(S). \tag{9}$$

It is observed that $X_g(S) \in [0, 1]$. Let a random variable $M_i = \sum_{j=1}^{i} (X_g(S) - \mu_X), \forall i \geq 1$, where $\mu = \mathbb{E}[X_g(S)]$. For a sequence of random variables $M_1, M_2, \ldots$ we have:

$$\mathbb{E}[M_i | M_1, \ldots, M_{j-1}] = \mathbb{E}[M_{i-1}] + \mathbb{E}[X_i(A) - \mu] \tag{10}$$

$$= \mathbb{E}[M_{i-1}]. \tag{11}$$

Therefore, $M_1, M_2, \ldots$ is a form of the martingale [54]. We utilize the following Lemma, which is trivially derived from the martingale theory in [54]:

Algorithm 1: Generating a GRR sample

Input: Social network $G = (V, E)$, set of groups $\mathcal{C} = \{C_1, C_2, \ldots, C_K\}$
Output: A GRR sample R_g

1: Randomly pick a source group C_i among $\mathcal{C}$
2: **for** *each node* $u \in C_i$ **do**
3: Initialize a queue $Q = \{u\}$ and $R_g(u) = \{u\}$
4: **while** $Q \neq \emptyset$ **do**
5: $v \leftarrow Q.pop()$
6: **foreach** $u \in N_{in}(v) \setminus (R_j \cup Q)$ **do**
7: **if** (u, v) *was visited* **then**
8: $Q.push(u)$, $R_j \leftarrow R_j \cup \{u\}$
9: **else**
10: With probability $p(u, v)$:
11: mark (u, v) is visited
12: $Q.push(u)$, $R_j \leftarrow R_j \cup \{u\}$
13: **end**
14: **end**
15: **end**
16: **end**
17: **return** $R_g = \{R_g(u) | u \in C_i\}$

Lemma 2 ([54]). *Given a set of MRR samples $\mathcal{R}$ with $T = |\mathcal{R}|$ and $\lambda > 0$, we have:*

$$\Pr\Big[\sum_{j=1}^{T} X_j(S) - T \cdot \mu \geq \lambda\Big] \leq e^{-\frac{\lambda^2}{\lambda \frac{2}{3} + 2\mu T}} \tag{12}$$

$$\Pr\Big[\sum_{j=1}^{T} Z_j(S) - T \cdot \mu \leq -\lambda\Big] \leq e^{-\frac{\lambda^2}{2\mu T}}. \tag{13}$$

In Lemma 2, by replacing $\lambda = \epsilon T \mu$ with a note that $\sigma(S) = K\mu$, we have:

$$\Pr[\hat{\sigma}(S) \geq (1 + \epsilon)\sigma(S)] \leq e^{-\frac{\epsilon^2 \mu T}{2 + \frac{2}{3}\epsilon}} \tag{14}$$

$$\Pr[\hat{\sigma}(S) \leq (1 - \epsilon)\sigma(S)] \leq e^{-\frac{\epsilon^2 \mu T}{2}}. \tag{15}$$

Therefore, if the number of samples is at least $T \geq (2 + \frac{2}{3}) \frac{1}{\mu} \frac{1}{\epsilon^2} \ln(\frac{1}{\delta})$ for $\delta \in (0, 1)$, $\hat{\sigma}_{\mathcal{R}}(S)$ is an (ϵ, δ)-approximation of $\sigma(S)$, i.e,

$$\Pr[(1 - \epsilon)\sigma(S) \leq \hat{\sigma}_{\mathcal{R}}(S) \leq (1 + \epsilon)\sigma(S)] \geq 1 - \delta. \tag{16}$$

The next section is going to use this observation for devising an algorithm that guarantees the estimation of the group influence function.

5. Proposed algorithms

This section proposes new algorithms for the GIM problem. From the analysis and dissolution in Section 4, one can use $\hat{\sigma}(S)$ to effectively estimate $\sigma(S)$ if the number of samples $|\mathcal{R}|$ is sufficiently large. Therefore, to overcome the computationally hard calculation of $\sigma(\cdot)$, it is an alternative to solve the following problem instead of solving GIM directly.

Definition 3 (*Samples Influence with Minimal Cost (*SIM*) Problem*)**.** Given a graph (G, C) as an instance of the GIM problem and $\mathcal{R}$ as a set of GRR samples, the problem aims at finding a set of nodes $S \subseteq V$ with minimal total cost so that $\hat{\sigma}(S) = K$, i.e. finding $S = \arg\min_{S' \subseteq V : \hat{\sigma}(S) = K} c(S')$.

To solve the GIM problem with theoretical bounds, we first generate a set of GRR samples $\mathcal{R}$ and propose the *Modified Greedy* (MoGreedy), an approximation algorithm for the SIM problem. We then find solutions of SIM for multiple sets of samples and select a final solution. We prove the approximation guarantees by utilizing the martingale theory [54].

5.1. A bi-criteria approximation algorithm

We first propose the MoGreedy, a $(1+\ln(|\mathcal{R}|t_{max}))\frac{b_{max}}{b_{min}}$-approximation algorithm for SIM and then use it as the core of the proposed bi-criteria approximation algorithm.

5.1.1. An approximation algorithm for SIM

First of all, it is not hard to prove that SIM problem also is **NP**-hard and $\hat{\sigma}_{\mathcal{R}}(\cdot)$ is non-submodular and non-supermodular. To develop an approximation algorithm with a theoretical bound, we first introduce a lower bound function F of $\hat{\sigma}_{\mathcal{R}}(\cdot)$ and exploit its properties.

Define $f(S, R_g)$ as the total score of all source nodes in R_g which are influenced by the set S in sample graph g:

$$f(S, R_g) = \sum_{u \in C(R_g)} \text{ScoreCover}(S, R_g(u)) \tag{17}$$

It is easy to see that $f(S, R_g)$ is a non-negative and non-increasing set function respect to $S \subseteq V$. Denote $\Delta_T f(S, R_g) = f(S \cup T, R_g) - f(S, R_g)$ for any subset $T \subseteq V$ and set function f. If $T = \{u\}$, we simplify $\Delta_{\{u\}} f(S, R_g)$ by $\Delta_u f(S, R_g)$. An essential property of $f(\cdot, R_g)$ is introduced below.

Lemma 3. *For all $S \subseteq T \subseteq V$ and $v \notin T$, we have:*

$$\Delta_v f(S, R_g) \geq \frac{b_{min}}{b_{max}} \cdot \Delta_v f(T, R_g). \tag{18}$$

where $b_{min} = \min_{v \in V} b(v)$, $b_{max} = \max_{v \in V} b(v)$.

Define a set function $g(S, R_g) = \min\left\{1, \frac{f(S,R_g)}{t(R_g)}\right\}$, we also have the following Lemma.

Lemma 4. *For all $S \subseteq T \subseteq V$ and $v \notin T$, we have:*

$$\Delta_v g(S, R_g) \geq \frac{b_{min}}{b_{max}} \cdot \Delta_v g(T, R_g) \tag{19}$$

In order to influence all samples in $\mathcal{R}$, the algorithm must find S such that $g(S, R_g) = 1, \forall R_g \in \mathcal{R}$. Therefore, it needs to find S with the minimal total cost such that

$$F(S, \mathcal{R}) = \frac{K}{T} \sum_{R_g \in \mathcal{R}} g(S, R_g) = \hat{\sigma}(S) = K. \tag{20}$$

Since $F(S, \mathcal{R})$ is a linear combinatorial of $g(S, R_g)$, it is easy to show the following inequality.

$$\Delta_v F(S, \mathcal{R}) \geq \frac{b_{min}}{b_{max}} \cdot \Delta_v F(T, \mathcal{R}) \tag{21}$$

for all $S \subseteq T \subseteq V$ and $v \notin V \setminus T$. $F(\cdot)$ is a lower bound function of $\hat{\sigma}_{\mathcal{R}}(\cdot)$ and they have the same value of the set S which can influence all samples in $\mathcal{R}$.

Description of MoGreedy. Based on the above theoretical analysis, we propose the MoGreedy algorithm (Algorithm 2), which utilizes the above characteristic of $F(\cdot, \mathcal{R})$. The algorithm adapts the idea of naive greedy [25], but it uses the value of F instead of f. In particular, S is initiated as empty, and then the algorithm iteratively adds a node v into the current solution S, which maximizes the marginal gain per its cost, i.e.,

$$\frac{\min\{K, F(S \cup \{v\}, \mathcal{R})\} - F(S, \mathcal{R})}{c(v)}$$

until the value of $F(S)$ achieves K.

Denoted by $S_i = \{s_1, s_2, \ldots, s_i\}$ the solution after i iterations in Algorithm 2, $S^0 = \{s_1^0, s_2^0, \ldots, s_k^0\}$ is an optimal solution of the SIM problem and let $\text{opt} = c(S^0)$. The following Lemma establishes a connection between the optimal and the candidate.

Algorithm 2: MoGreedy($\mathcal{R}, C$)

Input: A set of GRR samples $\mathcal{R}$, set of groups $C = \{C_1, C_2, \ldots, C_K\}$
Output: Seed set S

$S \leftarrow \emptyset$
while $F(S, \mathcal{R}) < K$ **do**
  $v_{max} \leftarrow \arg\max_{v \in V \setminus S} \frac{\min\{K, F(S \cup \{v\}, \mathcal{R})\} - F(S, \mathcal{R})}{c(v)}$
  $S \leftarrow S \cup \{v_{max}\}$
end
return S;

Lemma 5. *At the iteration i in the* MoGreedy *algorithm, we have:*

$$K - F'(S_i) \leq \text{opt} \cdot \frac{b_{max}}{b_{min}} \cdot \frac{F'(S_{i+1}) - F'(S_i)}{c(s_{i+1})} \tag{22}$$

where $F'(S) = \min\{K, F(S, \mathcal{R})\}$

Theorem 2. *Algorithm 2 provides a $\frac{b_{max}}{b_{min}}(1 + \ln(|\mathcal{R}|t_{max}))$-approximation solution for the* SIM *problem.*

Proof. Denote by $S_i = \{s_1, s_2, \ldots, s_i\}$ the solution of the algorithm after iteration i of the main loop. By using Lemma 5, we have:

$$\begin{aligned} K - F'(S_{i+1}) &\leq \left(1 - \frac{b_{min}}{b_{max}} \frac{c(s_{i+1})}{\text{opt}}\right)(K - F'(S_i)) \\ &\leq e^{-\frac{b_{min}}{b_{max}} \frac{c(s_{i+1})}{\text{opt}}} \cdot (K - F'(S_i)) \\ &\leq e^{-\frac{b_{min}}{b_{max}} \frac{\sum_{j=1}^{i+1} c(s_{i+1})}{\text{opt}}} \cdot K. \end{aligned}$$

It follows that

$$\begin{aligned} K - F'(S_{l-1}) &\leq e^{-\frac{b_{min}}{b_{max}} \frac{\sum_{j=1}^{l-1} c(s_j)}{\text{opt}}} \cdot K = e^{-\frac{b_{min}}{b_{max}} \frac{c(S_{l-1})}{\text{opt}}} \cdot K \\ \Longrightarrow c(S_{l-1}) &\leq \frac{b_{max}}{b_{min}} \text{opt} \cdot \ln \frac{K}{K - F'(S_{l-1})} \\ &\leq \frac{b_{max}}{b_{min}} \text{opt} \cdot \ln(|\mathcal{R}|t_{max}). \end{aligned}$$

The last inequality is due to:

$$K - F'(S_{l-1}) \geq K - F(S_{l-1}) \geq \frac{K}{|\mathcal{R}|} \frac{1}{t_{max}}.$$

Also, from Lemma 5 it implies that $c(s_l) \leq \frac{b_{max}}{b_{min}}\text{opt}$. Therefore:

$$c(S) = c(S_{l-1}) + c(s_l) \leq (1 + \ln(|\mathcal{R}|t_{max}))\frac{b_{max}}{b_{min}}\text{opt}.$$

We complete the proof. □

Complexity. At each iteration, MoGreedy scans at most n nodes and calculates the marginal gain value of F'. Therefore, it takes $O(|S|n)$ time complexity.

5.1.2. Bi-criteria approximation algorithm for GIM

We now present GIA, a randomized bi-criteria approximation. GIA returns a $(1-\epsilon, O(\ln K + \ln\ln n))$-bi criteria approximation solution w.h.p for the GIM problem. This algorithm is inspired by the idea of the Stop-and-Stare algorithm for the IM problem [22], which introduces a stopping condition to check the quality of several candidate solutions.

However, due to the difference between GIM and IM, we have to give another stopping condition to check the candidate solutions and establish the number of required samples that ensures the theoretical bounds of the final solution. Thus, the proposed approach completely differs from algorithms in [12], which inherits the Stop-and-Stare framework with some minor modifications.

Algorithm 3: GIA Algorithm

Input: Graph $G = (V, E)$, set of groups $C = \{C_1, C_2, \dots, C_K\}$, $\epsilon, \delta \in (0, 1)$.
Output: A seed set S

1: $N_{max} = (2 + \frac{2}{3}\epsilon)\frac{K}{\epsilon^2} \ln\left(\frac{2\binom{n}{k_{max}}}{\delta}\right)$, $N_1 \leftarrow (2 + \frac{2}{3}\epsilon)\frac{1}{\epsilon^2}\ln(\frac{1}{\delta})$
2: $i_{max} \leftarrow \lceil \log_2(N_{max}/N_1) \rceil$, $\delta_1 \leftarrow \frac{\delta}{2i_{max}}$
3: Generate set of N_1 samples $\mathcal{R}_1$
4: **for** $i = 1$ *to* i_{max} **do**
5: $S_i \leftarrow$ MoGreedy$(\mathcal{R}_i, C)$
6: Calculate $F_l(S, \mathcal{R}_i, \epsilon, \delta_1)$ by equation (23)
7: **if** $F_l(S, \mathcal{R}_i, \epsilon, \delta_1) \geq K - \epsilon K$ *or* $i = i_{max}$ **then**
8: **break**
9: **else**
10: Double size of $\mathcal{R}_i$ by generating $|\mathcal{R}_i|$ samples and adding them into $\mathcal{R}_i$
11: $\mathcal{R}_{i+1} \leftarrow \mathcal{R}_i$
12: **end**
13: **end**
14: **return** S

Description of GIA. The GIA algorithm receives an instance (G, C) and accuracy parameters ϵ, δ as inputs. It consists of several iterations and finds a candidate solution at each iteration by leveraging the MoGreedy algorithm and checks the quality of these solutions based on static evidence via Lemmas 6. Denote $k_{max} = \arg\max_{k=1\dots n}\binom{n}{k}$, the algorithm needs at most:

$$N_{max} = (2 + \frac{2}{3}\epsilon)\frac{K}{\epsilon^2}\ln(2\binom{n}{k_{max}}/\delta).$$

samples and needs at most $i_{max} = \lceil \log_2(N_{max}/N_1) \rceil$ iterations, where $N_1 = (2+\frac{2}{3}\epsilon)\frac{1}{\epsilon^2}\ln(\frac{n}{\delta})$. N_{max} is the number of samples required to ensure the approximation ratio, which is shown by Theorem 3.

At iteration i, the algorithm creates a set of $(2 + \frac{2}{3}\epsilon)\frac{1}{\epsilon^2}\ln(\frac{1}{\delta})2^{i-1}$ samples $\mathcal{R}_i$ and finds a candidate solution S_i by adapting MoGreedy algorithm (line 5). We devise a stopping condition and check the quality of S_i in line 7. Note that this algorithm does not reuse the stopping condition in [22], which is used in a recent work [12]. In this algorithm, we introduce the function $F_l(S, \mathcal{R}, \epsilon, \delta)$ of f, defined in Lemma 6 as a lower bound of F. This property is proved in Lemma 6. The stopping condition is essential to obtain the approximation ratio more succinct than the Stop-and-Stare in [22]. If the condition is satisfied, the algorithm returns S_i as a final solution. Otherwise, it doubles the size of $\mathcal{R}_i$ and moves to the next iteration. The details of GIA are presented in Algorithm 3.

Theoretical analysis. The approximation analysis is based on the martingale theory [54]. By applying Lemma 2, $F_l(S, \mathcal{R}, \epsilon, \delta)$ is a lower bound function of $\sigma(S)$ with high probability.

Lemma 6. *Given accuracy parameters $\epsilon, \delta \in (0, 1)$, a set $S \subseteq V$ and a set of GRR samples $\mathcal{R}$. Denote $c = \ln(1/\delta)$, $T = |\mathcal{R}|$ and*

$$F_l(S, \mathcal{R}, \epsilon, \delta) = \min\{\hat{\sigma}_{\mathcal{R}}(S) - \frac{Kc}{3T}, \hat{\sigma}_{\mathcal{R}}(S) + \frac{K}{T}(\frac{2c}{3} - \sqrt{\frac{4c^2}{9} + 2Tc\frac{\hat{\sigma}_{\mathcal{R}}(S)}{K}})\}. \quad (23)$$

We have $\Pr[\sigma(S) \geq F_l(S, \mathcal{R}, \epsilon, \delta)] \geq 1 - \delta$.

Lemma 7 shows an interesting property of the optimal solution of the GIM problem, which is to find a connection between the final solution and the optimal solution.

Lemma 7. *For any set of GRR samples $\mathcal{R}$, we have: $\hat{\sigma}_{\mathcal{R}}(S^*) = K$.*

The performance ratio of GIA algorithm is formally claimed in Theorem 3.

Theorem 3. *For any input parameters $\epsilon, \delta \in (0, 1)$,* GIA *algorithm returns a solution S such that $\Pr[\sigma(S) \geq K - \epsilon K] \geq 1 - \delta$ and $c(S) \leq \frac{b_{max}}{b_{min}}(1 + \ln\left((2 + \frac{2}{3}\epsilon)\epsilon^{-2}\right) + \ln K + \ln(nt_{max}\ln(n/\delta)))\mathsf{OPT}$.*

Proof. Denote $\mu = \frac{\sigma(S)}{K}, \hat{\mu} = \frac{\hat{\sigma}(S)}{K} = 1$ and $c = \ln(\binom{n}{k_{max}}/\delta)$. In Algorithm 3, we consider following bad events $B_i : \sigma(S_i) < K - \epsilon K$, for each iteration $i = 1, \dots, i_{max}$. Two following cases happen:

Case 1. If the algorithm terminates at some iterations $i = 1, \dots, i_{max} - 1$. Applying Lemma 6, we have:

$$\Pr(B_i) = \Pr[\sigma(S_i) < K - K\epsilon] \leq \Pr[\sigma(S_i) < F_l(S_i, \mathcal{R}_i, \epsilon, \delta_1)] \leq \delta_1.$$

Case 2. If the algorithm stops at iteration i_{max}, applying Lemma 6 with a note that $T = |\mathcal{R}| = (2 + \frac{2}{3}\epsilon)\frac{K}{\epsilon^2}\ln\left(2\frac{\binom{n}{k_{max}}}{\delta}\right) \geq 2c/\epsilon^2$ and $\hat{\mu} = 1$, the following event happens with a probability of at least: $1 - \frac{\delta}{2\binom{n}{k_{max}}}$:

$$\begin{aligned}
\mu &\geq \min\left\{\hat{\mu} - \frac{c}{3T}, \hat{\mu} + \frac{1}{T}\left(\frac{2c}{3} - \sqrt{\frac{4c^2}{9} + 2Tc\hat{\mu}}\right)\right\} \\
&= \min\left\{1 - \frac{c}{3T}, 1 + \frac{1}{T}\left(\frac{2c}{3} - \left(\frac{2c}{3} + \sqrt{2Tc}\right)\right)\right\} \\
&\quad \text{(Since } a^2 + b^2 \leq (a+b)^2, a, b > 0) \\
&\geq \min\left\{1 - \frac{\epsilon^2}{6}, 1 - \sqrt{\frac{2c}{T}}\right\} \\
&\geq \min\left\{1 - \frac{\epsilon^2}{6}, 1 - \epsilon\right\} \\
&\geq 1 - \epsilon.
\end{aligned}$$

Hence, $\Pr[B_{i_{max}}] = \Pr[\mu < 1 - \epsilon] \leq \frac{\delta}{2\binom{n}{k_{max}}}$. Assuming that $|S| = k$, there are at most $\binom{n}{k}$ possible solution, so we have:

$$\Pr[\forall S_{i_{max}} : B_{i_{max}}] \leq \binom{n}{k}\frac{\delta}{2\binom{n}{k_{max}}} \leq \frac{\delta}{2}.$$

By the union bound of the probabilities, none of the events B_i, $i = 1, \dots, i_{max}$ happens with a probability at least $1 - (i_{max}\delta_1 + \frac{\delta}{2}) \geq 1 - \delta$. It implies:

$$\Pr[\sigma(S) \geq K - \epsilon K] \geq 1 - \delta.$$

Denote $S_i^0 = \arg\min_{S:\sigma_{\mathcal{R}_i}(S)=K} c(S)$ and $\mathsf{opt}_i = c(S_i^0)$, where $\sigma_{\mathcal{R}_i}(S)$ is an estimation of $\sigma(S)$ over $\mathcal{R}_i$. From Lemma 7, we have $\sigma_{\mathcal{R}_i}(S^*) = K$, therefore $\mathsf{opt}_i \leq c(S^*)$. From Lemma 1, we have:

$$\begin{aligned}
c(S_i) &\leq \frac{b_{max}}{b_{min}} \cdot \left(1 + \ln(N_i t_{max})\right)\mathsf{opt}_i \\
&\leq \frac{b_{max}}{b_{min}} \cdot \left(1 + \ln(N_{max} t_{max})\right)\mathsf{opt}_i \\
&\leq \frac{b_{max}}{b_{min}}(1 + \ln\left((2 + \frac{2}{3}\epsilon)\epsilon^{-2}\right) + \ln K + \ln(nt_{max}\ln(n/\delta)))\mathsf{OPT}
\end{aligned}$$

which completes the proof. □

Theorem 4 (*Complexity*)**.** GIA *algorithm has*

$$O\left((n\ln n + \ln(\frac{1}{\delta})\epsilon^{-2})|C|\eta + n^2)\log n\right)$$

time complexity, where η is the expectation of influence spread of a node.

Proof. The complexity of the algorithm comes from generating GRR samples and running the MoGreedy algorithm. Denoted by $\mathbb{I}(S, v)$ the

probability that a node-set S influences v, and denoted by $\mathbb{I}(S)$ influence spread of node set S, we obtain:

$$\begin{aligned}\mathbb{E}[|R_g|] &= \frac{1}{K}\left(\sum_{C_i\in C}\sum_{v\in C_i}\sum_{u\in V}\mathbb{I}(\{u\},v)\right)\\ &= \frac{1}{K}\left(\sum_{v\in C}\sum_{u\in V}\mathbb{I}(\{u\},v)\right)\\ &= \frac{|C|}{K}\frac{1}{|C|}\left(\sum_{u\in C}\sum_{v\in V}\mathbb{I}(\{u\},v)\right)\\ &= \frac{|C|}{K}\frac{1}{|C|}\sum_{u\in C}\mathbb{I}(\{u\})\\ &= \frac{|C|}{K}\eta.\end{aligned}$$

This implies that generating samples takes at most $O(N_{max}\frac{|C|\eta}{K})$ and the running time at any iteration i is at most:

$$(k_{max}\ln n + \ln(\frac{1}{\delta})\epsilon^{-2})|C|\eta + |S_i|n = O\left((n\ln n + \ln(\frac{1}{\delta})\epsilon^{-2})|C|\eta + n^2\right).$$

On the other hand,

$$\begin{aligned}i_{max} &= O(\log\frac{N_{max}}{N_1}) = O(\log(Kn\log n))\\ &= O(\log K + \log n + \log\log n)\\ &= O(\log n)\quad(\text{Since } K\le n).\end{aligned}$$

Combining the above equalities, we obtain the time complexity of the algorithm. □

5.2. Exact groups influence algorithm

We further propose the EGI algorithm, an (almost) exact solution with high probability for GIM by using integer programming for solving the SIM problem instead of the MoGreedy algorithm and reusing the algorithmic framework of Algorithm 3.

Given a set of samples $\mathcal{R}$, we formulate the integer Linear Programming (IP) for solving the SIM problem for an instance $(\mathcal{R}, C)$ of the SIM problem, denoted by $\text{IP}(\mathcal{R}, C)$, as follows:

$$\text{min:}\quad \sum_{v\in v} x_v c(v) \tag{24}$$

$$\text{s.t:}\quad \sum_{u\in R_g}\min\left\{\sum_{v\in R_g(u)} x_v, 1\right\} b(u) \ge t(R_g), \qquad \forall R_g\in\mathcal{R} \tag{25}$$

$$x_v\in\{0,1\}, \qquad \forall v\in V \tag{26}$$

where

$$x_i = \begin{cases}1, \text{if } v \text{ is selected in the solution S}\\ 0, \text{otherwise.}\end{cases} \tag{27}$$

The objective of the IP is to select a seed set with minimal total cost. The constraints (25), (26) ensure all target groups be influenced by S.

The details of EGI are presented in Algorithm 4. MoGreedy in GIA is replaced by solving $\text{IP}(\mathcal{R}_i, C)$ (line 7). The rest of this algorithm is the same as EGI. By the same reasoning as that in Theorem 3, we can also prove the approximation ratio of EGI as follows.

Theorem 5. *For any $\epsilon, \delta \in (0,1)$, the Algorithm 4 returns a solution S such that $\sigma(S) \ge K - \epsilon K$ and $c(S) \le \text{OPT}$ with probability at least $1-\delta$.*

Proof. Similar to the Proof of Theorem 3, we also have $\hat{\sigma}(S) = K$ and $\Pr[\sigma(S) \ge K - \epsilon K] \ge 1-\delta$. On the other hand, solving the $\text{IP}(\mathcal{R}_i, C)$ provides an optimal solution for the SIM problem. From Lemma 7, we have $c(S_i) = \text{opt}_i \le \text{OPT}$, $\forall i = 1 \dots i_{max}$. Therefore $c(S) \le \text{OPT}$. □

Algorithm 4: EGI Algorithm

Input: Graph $G = (V, E)$, set of K target groups $C = \{C_1, C_2, \dots, C_K\}$, $\epsilon, \delta \in (0,1)$.
Output: A Seed set S

1. $N_{max} = (2 + \frac{2}{3}\epsilon)\frac{K}{\epsilon^2}\ln\left(\frac{2\binom{n}{k_{max}}}{\delta}\right)$, $N_1 \leftarrow \frac{1}{\epsilon^2}\ln(\frac{1}{\delta})$
2. $i_{max} \leftarrow \lceil\log_2(N_{max}/N_1)\rceil$, $\delta_1 \leftarrow \frac{\delta}{2(i_{max}-1)}$
3. Generate set of N_1 samples $\mathcal{R}_1$
4. **for** $i = 1$ *to* i_{max} **do**
5. $\quad S_i \leftarrow$ a solution by solving $\text{IP}(\mathcal{R}_i, C)$.
6. $\quad$ Calculate $F_l(S_i, \mathcal{R}_i, \epsilon, \delta_1)$ by Lemma 6
7. $\quad$ Double size of $\mathcal{R}_i$ by generating $|\mathcal{R}_i|$ samples and add them into $\mathcal{R}_i$
8. $\quad \mathcal{R}_{i+1} \leftarrow \mathcal{R}_i$
9. $\quad$ **if** $F_l(S_i, \mathcal{R}_{i+1}, \epsilon, \delta_1) \ge (1-\epsilon)K$ *or* $i = i_{max}$ **then**
10. $\quad\quad$ **return** S_i
11. $\quad$ **end**
12. **end**
13. **return** S_i

Table 2
Datasets

Dataset	Nodes	Edges	Type	Avg.degree
Facebook	747	60.05K	Directed	81
Wiki	7.1K	103.6K	Directed	15
Epinions	76K	508.8K	Directed	7
DBLP	317K	1.05M	Directed	4
Pokec	1.6M	30.6M	Directed	20

6. Experiments

This section shows the proposed algorithms' performance is illustrated by conducting comprehensive experiments. We compared ours with the state-of-the-art algorithms on two major metrics: *(1) solution quality* and *(2) computing time* on various network datasets.

6.1. Experimental settings

Dataset. We use some public OSN datasets in the experiments, shown in Table 2. These data sets are widely used in the related work [12,55].

Parameters setting. All experiments are under the IC model with edge probabilities set to $p(u,v) = 1/|N_{in}(v)|$. This weight setting is adopted from prior works [1,2,12,20–23]. We set parameters $\epsilon = 0.1$, $\delta = 1/n$ and the limited time is 6 h. For the purpose of providing a comprehensive experiment, we divide the experiment into the following two cases.

- **Case 1. Uniform Cost (UC).** In this case, $s(u) = 1, \forall u \in U$, and the thresholds $t_i = \sum_{u\in C_i} s(u)/2$ for $i = 1 \dots K$ according to the setting in [12] and the cost $c(u) = 1, \forall u \in V$.
- **Case 2. General Cost (GC).** Each node has its cost calculated under the Normalized Linear model with the support $(0,1]$ according to recent works [38,56,57] and $s(u) = 1, \forall u \in V$, and the thresholds $t_i = \sum_{u\in C_i} s(u)/2$ for $i = 1 \dots K$ according to the setting in [12].

Algorithms compared. To our knowledge, there are no existing algorithms that can be adopted to solve the GIM problem directly. Therefore, we compare our GIA and EGI algorithms with the state-of-the-art algorithms for the closest problem: IMC [12]. Besides, we use High Degree, a common baseline algorithm for the related problem on information diffusion [1,18,28]. These algorithms are described in detail as follows.

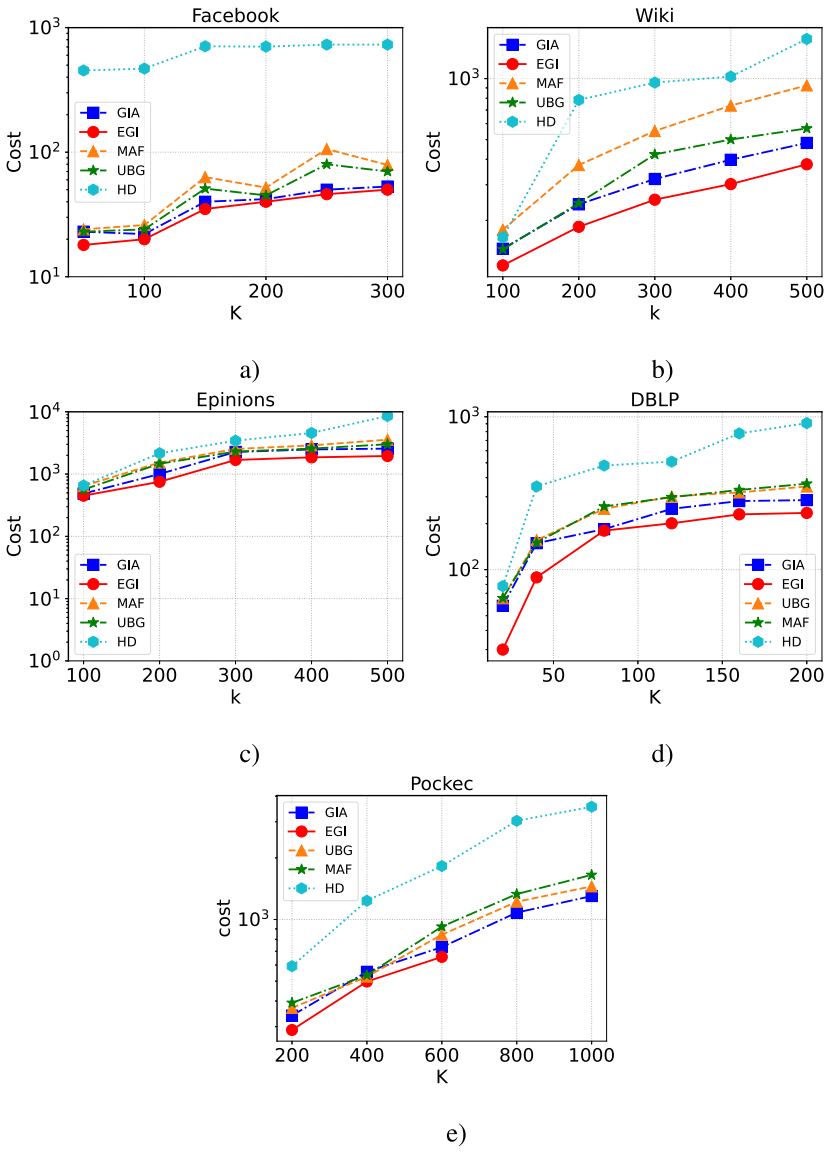

Fig. 1. Size of seed set returned by algorithms under the UC setting.

- **Upper Bound Greedy (UBG)** [12]. This is the best performance algorithm for the IMC problem. This problem asks to find a set seed of k nodes such that the number of influenced groups is maximal, while the GIM problem asks to find a set of seed nodes with minimal cost that can influence all target groups. Therefore, we adapt the UBG algorithm for solving GIM with some modifications as follows. We first initialize an empty candidate solution S. We then sequentially use UBG with k from 1 to n to find the best influence node and then add it into S until the estimation $\hat{\sigma}(S)$ is at least $(1-\epsilon)K$.
- **Most Appearance First (MAF)** [12]. This is also an algorithm for the IMC problem. We also adapt it as the workflow in UBG for the GIM problem.
- **High Degree (HD)**. We repetitively select a node with the highest degree until the current solutions influence all target groups.

For all the above algorithms, we use the Monte-Carlo method in [58] to obtain an (ϵ, δ)-approximation for estimating the influence group function. We implement GIA in C++ using CPLEX to solve the IP.

For each algorithm, we run five times to get the average results.

6.2. Experiment results

In this section, we show the experimental results and compare the algorithms based on two criteria: **(1) solution quality**, which is measured by the cost and the group function value of obtained solutions, and **(2) computational time.**

Solution quality. We first compare the solution quality of the algorithms under the UC case (Figs. 1 and 2). In this case, the cost of a solution is its size. GIA and EGI outperform others by a large gap. Specifically, they are up to 2.5 times better than UBG and MAF. EGI provides the smallest-cost solutions, which are up to approximately 1.2 times smaller than that of EGI (on Wiki). These results conses with the theoretical performance of EGI, that is, EGI returns the best approximation algorithm ratio for GIM problem. Although UBG can give better results than MAF and HD in general, it does not give any approximation ratio for the GIM problem. The selection of a fixed-size seed set in each iteration of the binary search causes the selection of many unnecessary seed nodes by UBG. The same issue confronts MAF. HD returns poor results since it is a simple heuristic and only considers the degree of nodes instead of having an influence on groups. The solution quality of the algorithms under GC is shown in Figs. 4 and 5. We first compare the total cost of seed sets. Similar to the previous case,

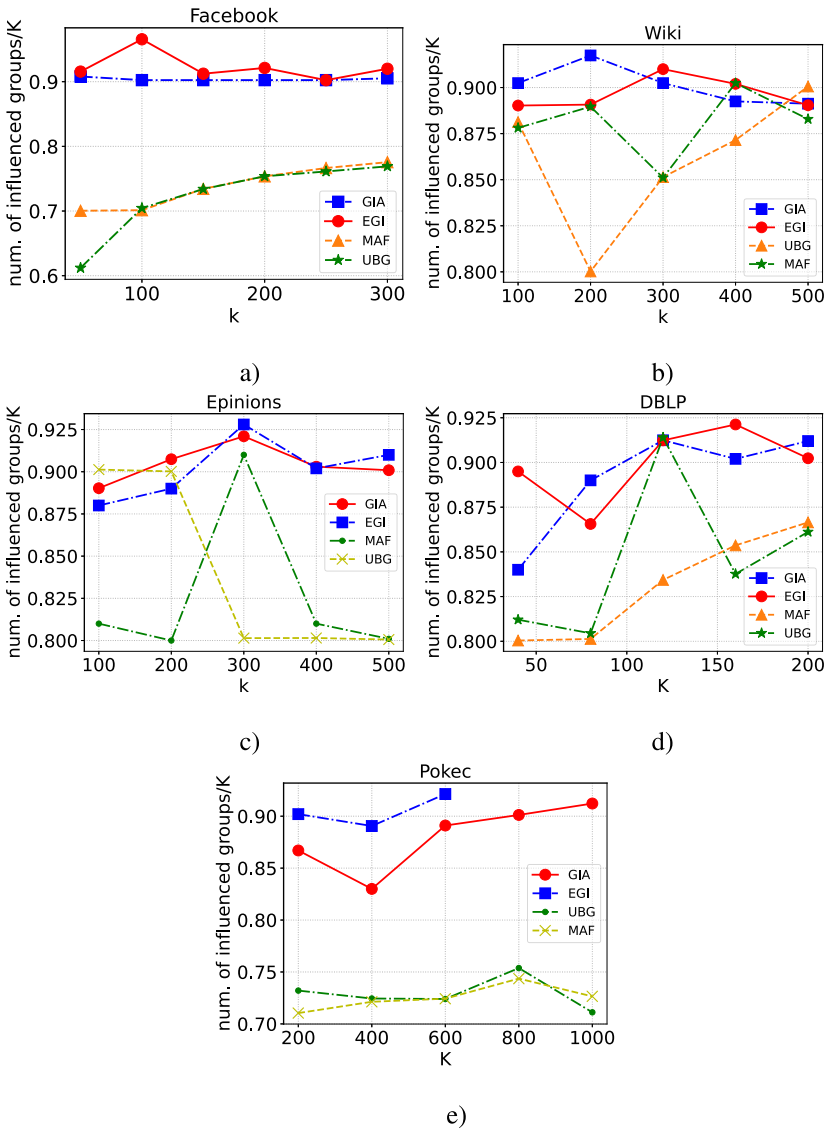

Fig. 2. Ratio of number of influenced groups over K of algorithms under the UC setting.

our algorithms outperform the others in terms of solution quality, and EGI also provides the best solution. It can be explained that MAF and UBG only consider the candidate seed sets with fixed sizes, and they fail to consider the cost of nodes. Again, our algorithms give ratios that are above $(1-\epsilon)$ in most cases.

We further show the group influence value of the algorithms. For the convenience of comparison with the value of K, we report the *ratio of number of influenced groups over* K of the algorithms in Figs. 2 and 5. It can be seen that GIA and EGI can output ratios that are above $(1-\epsilon)$ in most cases and outperform MAF and UBG. MAF and UBG give lower and unstable ratios even though they use a large number of samples (MAF and UBG use the RIC sample). This is due to two reasons: (1) our algorithms always make sure all GRRs are influenced, and (2) stopping conditions in our algorithms ensure that $\sigma(S) \geq (1-\epsilon)K$ with high certainty. These results confirm that the proposed algorithms provide a better solution than the other ones in both theoretical bounds and practice.

Computational time. We now compare the complexity of algorithms in practice via their computational time shown in Figs. 3 and 6. We do not report the running time of HD because it is a simple heuristic algorithm with a poor-quality solution and can finish within a few seconds. It is obvious that GIA is the fastest algorithm and outperforms the others by a huge distance. Particularly, it is up to approximately **840 and 646 times faster than MAF and UBG**, respectively.

Three possible explanations for this phenomenon are given: (1) the operation of MAF and UBG consists of many iterations to find the seed set that cannot reach the terminal condition timely. They use a binary search method to find feasible solutions and choose the best solution; (2) MAF and UBG use too many number of RIC samples to obtain an estimation of group influence function; (3) Our GIA follows the mechanism of our framework, which can find the final solution after a few iterations of the main loop. Our GIA can also apply for the large-scale network (Pocke with 1.6M nodes) within only a few seconds, confirming that the algorithmic framework and sampling technique is more efficient in the GIM problem than MAF and UBG. Our EGI has the longest running time since it uses the IP solver to find the exact solution for the sub-problem of SIM instead of the Modified Greedy. EGI gives the best quality of the solution. However, EGI takes some hours to solve GIM for small or medium networks, and EGI cannot be completed in the Pocke network when K is large within a limited time. Interestingly, even when K increases, the running time of our algorithms does not decrease in some cases. Our explanation is: the larger the value of K, the smaller number of samples needed for satisfying the terminal condition, enabling our algorithms to finish within fewer iterations. We

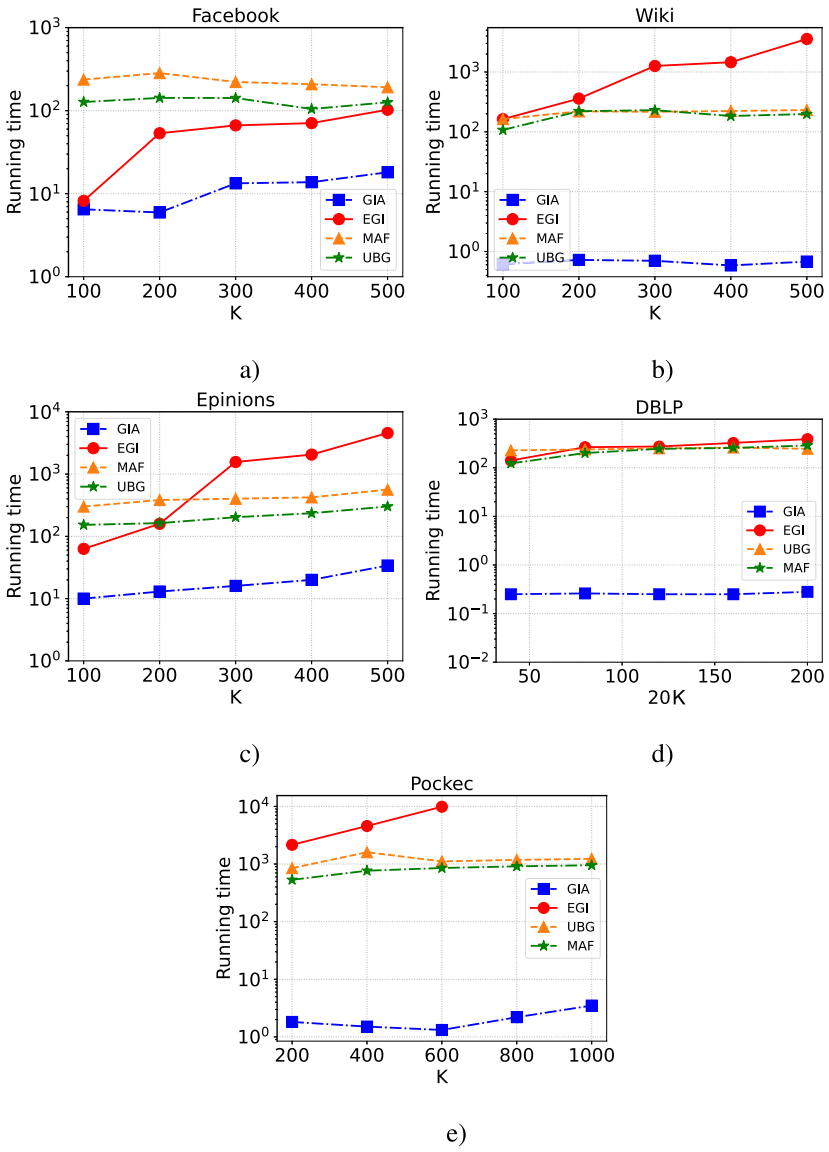

Fig. 3. Running time of algorithms under the UC setting.

also report the running time of the algorithms under GC case in Fig. 6. The results are consistent with the UC case: GIA is the fastest algorithm and up to approximately 800 times faster than MAF and UBG, and EGI has the longest running time.

7. Conclusions and discussions

In this paper, we investigated a novel GIM problem as follows: Given a social network and a set of target groups of users, the GIM problem aims to find a seed set with the smallest cost that can influence all groups. The studied problem arises from the goal of reaping the benefits of influencing user groups on social networks under a more realistic scenario with existing studies. This problem has a wide range applications including viral marketing, combating misinformation or fake news and defensing with bad effects (virus, malicious) in a network, etc. The challenges of solving lines in three points: (1) GIM is computationally hard, and it cannot be approximated with a log factor, (2) the group influence function is neither submodular nor supermodular. Therefore it does not admit existing greedy methods with a theoretical bound, and (3) calculating the group influence function is #P-Hard, so it cannot be calculated exactly in polynomial time.

In order to overcome those challenges of the problem, we proposed two efficient algorithms, GIA and EGI, with theoretical bounds. The key of these algorithms lies in two aspects: Firstly, a new concept of sampling was developed to estimate the group influence function effectively. The new sampling technique was a generalization of the RIC sample concept in [12]. Secondly, we proposed an algorithm framework that operated in multiple iterators in which each constructed a candidate solution by checking the quality via the new sampling technique. In general, our algorithms can be applied to minimization cost problems with non-submodular utility functions. Several comprehensive experiments on real social network datasets were conducted to compare our proposed with state-of-the-art. The results demonstrated that our algorithms outperformed the state-of-the-art ones in terms of total cost, and our GIA algorithm was up to about 800 faster than competitive algorithms. The results also showed the efficiency of our sampling technique and algorithmic framework, which not only provided theoretical bounds but also outperformed existing algorithms in the theory in practice. Besides, our algorithms can be applied in any network to deal with the attacking of bad effects (e.g., misinformation, vulnerable networks of nodes, viruses, fake news, etc.) by finding a set of nodes that has a strong influence on groups and monitor and protect them under the attack.

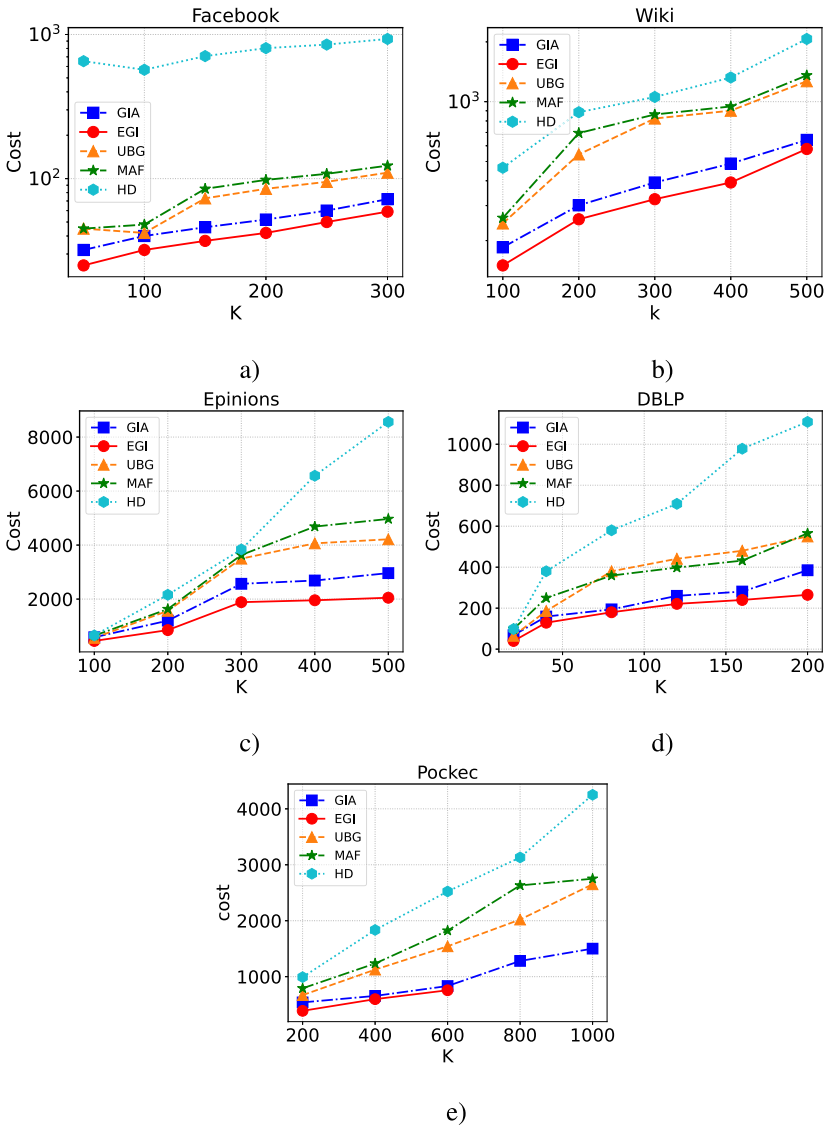

Fig. 4. Total cost of seed sets returned by algorithms under the GC setting.

There is still an open question for our algorithms, however: *How will our algorithms work on other information diffusion models?* In the future, we are going to investigate our algorithms under some other information diffusion models on both the theory and practice sides.

CRediT authorship contribution statement

Phuong N.H. Pham: Conceived and designed the analysis, Performed the analysis, Writing – original draft. **Canh V. Pham:** Conceived and designed the analysis, Contributed data or analysis tools, Performed the analysis, Writing – original draft. **Hieu V. Duong:** Contributed data or analysis tools, Performed the analysis. **Václav Snášel:** Conceived and designed the analysis, Performed the analysis. **Nguyen Trung Thanh:** Writing – original draft.

Declaration of competing interest

There is no conflict of interest in this work.

Data availability

Data will be made available on request.

Acknowledgment

This work was partially supported by the Ho Chi Minh City University of Industry and Trade - Vietnam through the HUIT fund for Science and Technology under Contract No. 84/HD-DCT. The second and the fifth authors would like to thank Avanced Study in Mathematics (VIASM) for their hospitality and financial support.

Appendix

This section provides missing proofs in Sections 4 and 5.

Proof of Lemma 1. We have

$$\sigma(S) = \sum_{C_i \in C} \sum_{g \sim G} \Pr[g \sim G] X_g(S, C_i) \tag{28}$$

$$= K \sum_{C_i \in C} \left(\frac{1}{K} \sum_{g \sim G} \Pr[g \sim G] X_g(S, C_i) \right) \tag{29}$$

$$= \sum_{C_i \in C} \sum_{g \sim G} \left(\Pr[C_i \text{ is a source group}] \Pr[g \sim G] X_g(S, C_i) \right) \tag{30}$$

$$= K \cdot \mathbb{E}[X_g(S)], \tag{31}$$

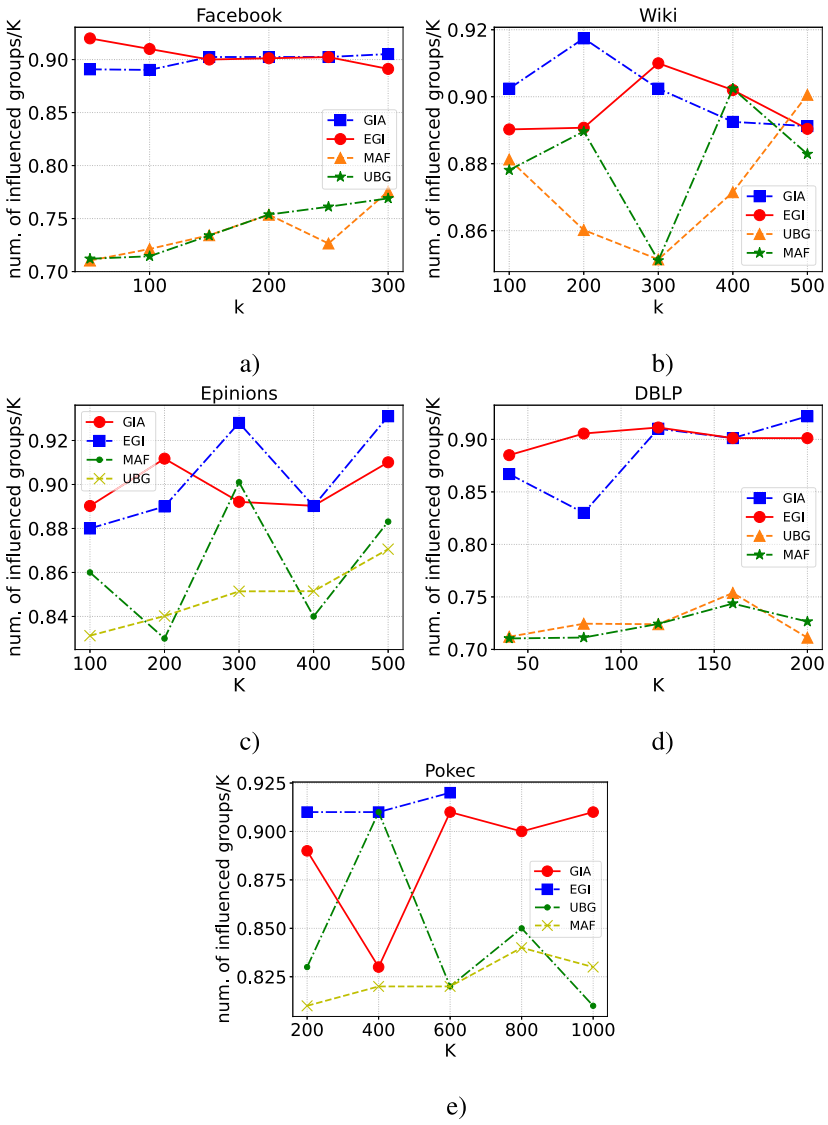

Fig. 5. Ratio of number of influenced groups over K of algorithms under the GC setting.

where $X_g(S, C_i)$ is the variable $X_g(S)$ with source group C_i. The Eq. (28) is due to the definition of $\sigma(S)$, and the Eq. (30) is due to the probability section of a source node. □

Proof of Lemma 3. If $v \notin R_g(u), \forall u \in C(R_g)$, the Lemma holds because $f(S \cup \{v\}, R_g) = f(S, R_g) = 0$ and $f(T \cup \{v\}, R_g) = f(T, R_g) = 0$.

If $\exists u \in C(R_g) : v \in R_g(u)$, we consider the following two sub-cases:
Case 1. If $\forall u \in C(R_g) : v \in R_g(u), T \cap R_g(u) \neq \emptyset$, the Lemma holds since $f(T \cup \{v\}, R_g) = f(T, R_g)$.
Case 2. If $\exists u \in C(R_g) : v \in R_g(u)$ and $T \cap R_g(u) = \emptyset$, we also have $S \cap R_g(u) = \emptyset$. It implies:

$$
\begin{aligned}
&f(S \cup \{v\}, R_g) - f(S, R_g) \\
&\geq b_{min} = \frac{b_{min}}{b_{max}} b_{max} \\
&\geq \frac{b_{min}}{b_{max}} \left(f(T \cup \{v\}, R_g) - f(T, R_g) \right)
\end{aligned}
$$

This completes the proof. □

Proof of Lemma 4. Since $f(\cdot, R_g)$ is non-decreasing so $f(S \cup \{v\}, R_g) \geq f(S, R_g)$ for any subset $S \subseteq$ and $v \in V$. Therefore, we only consider three possible following cases:

Case 1. $t(R_g) \geq f(S \cup \{v\}, R_g) \geq f(S, R_g)$, we have:

$$
\begin{aligned}
\Delta_v g(S, R_g) &= \min \left\{ 1, \frac{f(S \cup \{v\}, R_g)}{t(R_g)} \right\} - \min \left\{ 1, \frac{f(S, R_g)}{t(R_g)} \right\} \\
&= \frac{f(S \cup \{v\}, R_g) - f(S, R_g)}{t(R_g)}
\end{aligned}
$$

In this case, we further consider the following three sub-cases:

- If $t(R_g) \geq f(T \cup \{v\}, R_g) \geq f(T, R_g)$ then $g(T \cup \{v\}, R_g) = \frac{f(T \cup \{v\}, R_g)}{t(R_g)}$, and $g(T, R_g) = \frac{f(T, R_g)}{t(R_g)}$.
- If $f(T \cup \{v\}, R_g) \geq t(R_g) \geq f(T, R_g)$, then $g(T \cup \{v\}, R_g) = 1$, and $g(T, R_g) = \frac{f(T, R_g)}{t(R_g)}$.
- If $f(T \cup \{v\}, R_g) \geq f(T, R_g) \geq t(R_g)$, $g(T \cup \{u\}, R_g) = g(T, R_g) = 1$.

In the three cases above, we also have:

$$
\begin{aligned}
\Delta_v g(T, R_g) &= g(T \cup \{v\}, R_g) - g(T, R_g) \\
&\leq \frac{f(T \cup \{v\}, R_g) - f(T, R_g)}{t(R_g)} \\
&\leq \frac{b_{max}}{b_{min}} \frac{f(S \cup \{v\}, R_g) - f(S, R_g)}{t(R_g)} \text{ (By Lemma 3)}
\end{aligned}
$$

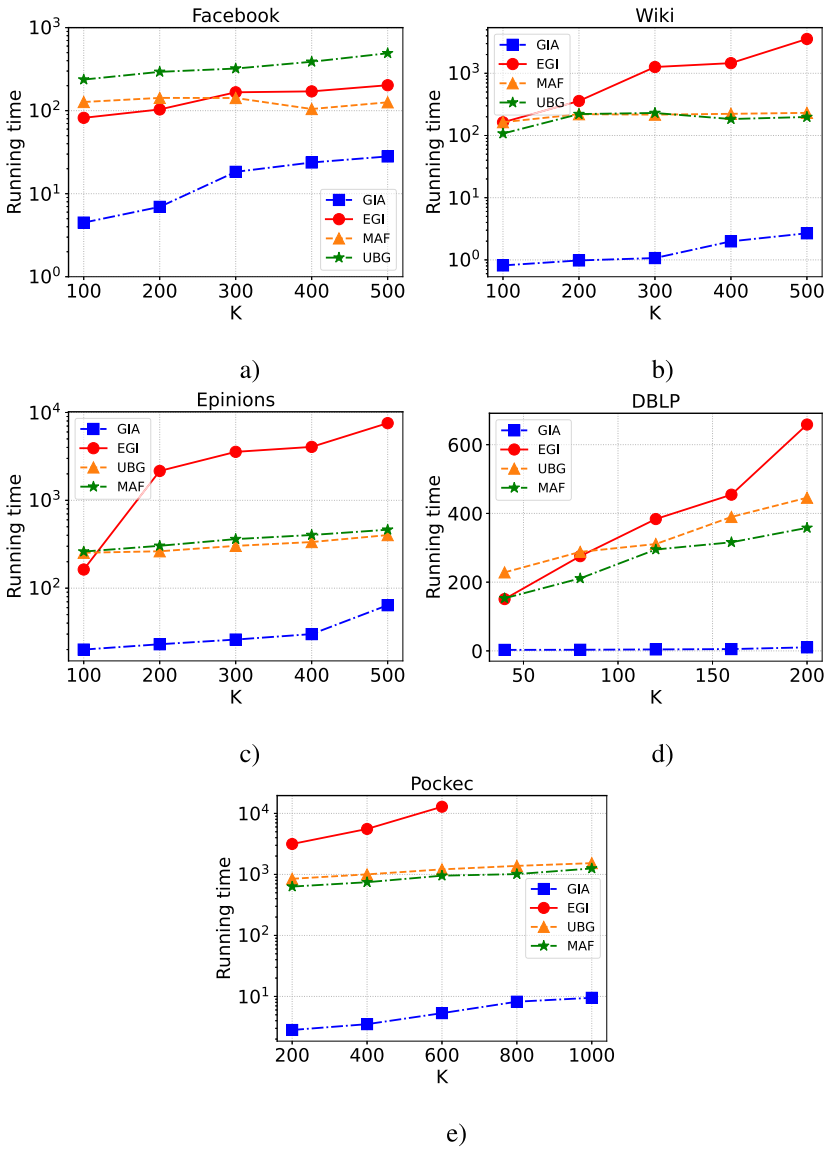

Fig. 6. Running time (second) of algorithms under the GC setting.

Case 2. $f(S \cup \{v\}, R_g) \geq t(R_g) \geq f(S, R_g)$. In this case, we have $f(T \cup \{v\}, R_g) \geq f(S \cup \{v\}, R_g) \geq t(R_g)$, and $g(S, R_g) = \frac{f(S,R_g)}{t(R_g)} \leq g(T, R_g) \leq 1$. Therefore $g(T \cup \{v\}, R_g) = g(S \cup \{v\}, R_g) = 1$, and

$$\begin{aligned}\Delta_v g(S, R_g) &= 1 - g(S, R_g) = g(T \cup \{v\}, R_g) - g(S, R_g) \\ &\geq g(T \cup \{v\}, R_g) - g(T, R_g) \\ &\geq \frac{b_{min}}{b_{max}} \cdot \left(g(T \cup \{v\}, R_g) - g(T, R_g)\right).\end{aligned}$$

Case 3. $f(S \cup \{v\}, R_g) \geq f(S, R_g) \geq t(R_g)$. It obtains:

$$\min\left\{1, \frac{f(S \cup \{v\}, R_g)}{t(R_g)}\right\} = \min\left\{1, \frac{f(S, R_g)}{t(R_g)}\right\} = 1$$
$$\min\left\{1, \frac{f(T \cup \{v\}, R_g)}{t(R_g)}\right\} = \min\left\{1, \frac{f(T, R_g)}{t(R_g)}\right\} = 1.$$

Therefore, $\Delta_v g(S, R_g) = \frac{b_{min}}{b_{max}} \Delta_v g(T, R_g) = 0$. The proof is proved. □

Proof of Lemma 5. It is easy to see that $\Delta_u F'(S) \geq \frac{b_{min}}{b_{max}} \cdot \Delta_u F'(T)$, for $S \subseteq T \subseteq V$ and $u \in V \setminus T$. Let $S' = S^0 \setminus S_i = \{s'_1, s'_2, \ldots, s'_t\}$, $S'_j = \{s'_1, s'_2, \ldots, s'_j\}, j \leq t$, and $S'_0 = \emptyset$, we have:

$$\begin{aligned}K - F'(S_i) &= F(S^0) - F'(S_i) \leq F'(S^0 \cup S_i) - F'(S_i) \\ &= F'(S_i \cup S') - F'(S_i) \\ &= \sum_{j=1}^{t} (F'(S_i \cup S'_j) - F'(S_i \cup S'_{j-1})) \\ &\leq \sum_{j=1}^{t} \frac{b_{max}}{b_{min}} (F'(S_i \cup s'_j) - F'(S_i)) \\ &\leq \frac{b_{max}}{b_{min}} \mathsf{opt} \cdot \frac{1}{c(S')} \sum_{j=1}^{t} (F'(S_i \cup s'_j) - F'(S_i)) \\ &\quad \text{(due to } c(S') \leq \mathsf{opt}) \\ &= \frac{b_{max}}{b_{min}} \mathsf{opt} \cdot \frac{\sum_{j=1}^{t} (F'(S_i \cup s'_j) - F'(S_i))}{\sum_{j=1}^{t} c(s'_j)}.\end{aligned}$$

For any positive numbers $a_1, \ldots, a_l$ and $b_1, \ldots, b_l$. According to [59], we have:

$$\min_{i=1\ldots l} \frac{a_i}{b_i} \leq \frac{\sum_{i=1}^{l} a_i}{\sum_{i=1}^{l} b_i} \leq \max_{i=1\ldots l} \frac{a_i}{b_i}. \tag{32}$$

Apply the above inequality, we obtain:

$$K - F'(S_i) \le \frac{b_{max}}{b_{min}}\mathsf{opt} \cdot \max_{s'_j \in S'} \frac{F'(S_i \cup s'_j) - F'(S_i)}{c(s'_j)} \le \frac{b_{max}}{b_{min}}\mathsf{opt} \cdot \frac{F'(S_i \cup s_{i+1}) - F'(S_i)}{c(s'_j)} \le \frac{b_{max}}{b_{min}}\mathsf{opt} \cdot \frac{F'(S_{i+1}) - F'(S_i)}{c(s_{i+1})}.$$

By rearranging the last inequality, the proof is completed. □

Proof of Lemma 6. Denote $\mu = \frac{\sigma(S)}{K}, \hat{\mu} = \frac{\hat{\sigma}(S)}{K}$ and $c = \ln(1/\delta)$. Apply (12) in Lemma 2 with $\lambda = \frac{c}{3} + \sqrt{\frac{c^2}{9} + 2c\mu T}$, we have:

$$\Pr\left[\sum_{j=1}^{T} X_j(S) - T \cdot \mu \ge \lambda\right] \le \delta \tag{33}$$

Therefore, the following event happens with probability at least $1 - \delta$:

$$\sum_{j=1}^{T} X_j(S) - T \cdot \mu \le \lambda \tag{34}$$

$$\iff \quad T\hat{\mu} - T\mu - \frac{c}{3} \le \sqrt{\frac{c^2}{9} + 2c\mu T}. \tag{35}$$

By solving the above inequality for finding μ, we have:

$$\mu \ge \min\left\{\hat{\mu} - \frac{c}{3T}, \hat{\mu} + \frac{1}{T}\left(\frac{2c}{3} - \sqrt{\frac{4c^2}{9} + 2Tc\hat{\mu}}\right).\right\} \tag{36}$$

Replace $\hat{\mu} = \frac{\hat{\sigma}(S)}{K}, \mu = \frac{\sigma(S)}{K}$ into above inequality, we obtain the proof. □

Proof of Lemma Lemma 7. We prove this Lemma by the contradiction hypothesis. Assuming that there exists a set of GRR samples $\mathcal{R}$ that $\hat{\sigma}_{\mathcal{R}}(S^*) < K$, i.e. there exists a set $\mathcal{R}_1 \subseteq \mathcal{R}$ so that $\sum_{R_g \in \mathcal{R}_1} X_g(S^*) = 0$.

Denote by Ω the space of GRR samples with a probability of generating a sample defined in Eq. (8), we have:

$$\sigma(S^*) = K \cdot \mathbb{E}[X_g(S^*)] = K \cdot \sum_{R_g \in \Omega} \Pr[R_g] X_g(S^*) = K \cdot \sum_{R_g \in \mathcal{R}_1} \Pr[R_g] X_g(S^*) + K \sum_{R_g \in \Omega \setminus \mathcal{R}_1} \Pr[R_g] X_g(S^*) = K \cdot \sum_{R_g \in \Omega \setminus \mathcal{R}_1} \Pr[R_g] X_g(S^*) = K \cdot \sum_{R_g \in \Omega \setminus \mathcal{R}_1} \Pr[R_g] < K.$$

The last inequality contracts with the fact that S^* is an optimal solution to the GIM problem. Therefore, $\hat{\sigma}_{\mathcal{R}}(S^*) = K$. □

References

[1] D. Kempe, J.M. Kleinberg, É. Tardos, Maximizing the spread of influence through a social network, in: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 24 - 27, 2003, 2003, pp. 137–146.

[2] H.T. Nguyen, M.T. Thai, T.N. Dinh, A billion-scale approximation algorithm for maximizing benefit in viral marketing, IEEE/ACM Trans. Netw. 25 (4) (2017) 2419–2429.

[3] Y. Li, D. Zhang, K. Tan, Real-time targeted influence maximization for online advertisements, Proc. VLDB Endow. 8 (10) (2015) 1070–1081.

[4] H. Zhang, M.A. Alim, X. Li, M.T. Thai, H.T. Nguyen, Misinformation in online social networks: Detect them all with a limited budget, ACM Trans. Inf. Syst. 34 (3) (2016) 18:1–18:24.

[5] H. Zhang, A. Kuhnle, H. Zhang, M.T. Thai, Detecting misinformation in online social networks before it is too late, in: Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM, San Francisco, CA, USA, August 18-21, 2016, 2016, pp. 541–548.

[6] C.V. Pham, D.V. Pham, B.Q. Bui, A.V. Nguyen, Minimum budget for misinformation detection in online social networks with provable guarantees, Optim. Lett. 16 (2) (2022) 515–544.

[7] C.V. Pham, M.T. Thai, H.V. Duong, B.Q. Bui, H.X. Hoang, Maximizing misinformation restriction within time and budget constraints, J. Comb. Optim. 35 (4) (2018) 1202–1240.

[8] 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, WWW 2011, Hyderabad, India, March 28 - April 1, 2011, 2011, pp. 665–674.

[9] H.T. Nguyen, A. Cano, V. Tam, T.N. Dinh, Blocking self-avoiding walks stops cyber-epidemics: A scalable GPU-based approach, IEEE Trans. Knowl. Data Eng. 32 (7) (2020) 1263–1275.

[10] C.V. Pham, Q.V. Phu, H.X. Hoang, J. Pei, M.T. Thai, Minimum budget for misinformation blocking in online social networks, J. Comb. Optim. 38 (4) (2019) 1101–1127.

[11] M. Ye, X. Liu, W. Lee, Exploring social influence for recommendation: a generative model approach, in: Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '12, Portland, OR, USA, August 12-16, 2012, 2012, pp. 671–680.

[12] L.N. Nguyen, K. Zhou, M.T. Thai, Influence maximization at community level: A new challenge with non-submodularity, in: Proceedings of the 39th IEEE International Conference on Distributed Computing Systems, ICDCS 2019, Dallas, TX, USA, July 7-10, 2019, 2019, pp. 327–337.

[13] A. Tsang, B. Wilder, E. Rice, M. Tambe, Y. Zick, Group-fairness in influence maximization, in: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019, 2019, pp. 5997–6005.

[14] J. Zhu, S. Ghosh, W. Wu, C. Gao, Profit maximization under group influence model in social networks, in: Computational Data and Social Networks - Proceedings of the 8th International Conference, CSoNet 2019, Ho Chi Minh City, Vietnam, November 18-20, 2019, 2019, pp. 108–119.

[15] G. Farnadi, B. Babaki, M. Gendreau, A unifying framework for fairness-aware influence maximization, in: Companion Proceedings of the Web Conference, Taipei, Taiwan, April 20-24, 2020, 2020, pp. 714–722.

[16] P.N.H. Pham, C.V. Pham, H.V. Duong, T.T. Nguyen, M.T. Thai, Groups influence with minimum cost in social networks, in: D. Mohaisen, R. Jin (Eds.), Computational Data and Social Networks - Proceedings of the 10th International Conference, CSoNet 2021, Virtual Event, November 15-17, 2021, in: Lecture Notes in Computer Science, 13116, Springer, 2021, pp. 231–242.

[17] W. Chen, C. Wang, Y. Wang, Scalable influence maximization for prevalent viral marketing in large-scale social networks, in: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '10, Association for Computing Machinery, New York, NY, USA, 2010, pp. 1029–1038.

[18] W. Chen, Y. Yuan, L. Zhang, Scalable influence maximization in social networks under the linear threshold model, in: ICDM 2010, Proceedings of the 10th IEEE International Conference on Data Mining, Sydney, Australia, 14-17 December 2010, 2010, pp. 88–97.

[19] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J.M. VanBriesen, N.S. Glance, Cost-effective outbreak detection in networks, in: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Jose, California, USA, August 12-15, 2007, 2007, pp. 420–429.

[20] Y. Tang, X. Xiao, Y. Shi, Influence maximization: near-optimal time complexity meets practical efficiency, in: Proceedings of the International Conference on Management of Data, SIGMOD 2014, Snowbird, UT, USA, June 22-27, 2014, 2014, pp. 75–86.

[21] Y. Tang, Y. Shi, X. Xiao, Influence maximization in near-linear time: A martingale approach, in: Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, Melbourne, Victoria, Australia, May 31 - June 4, 2015, 2015, pp. 1539–1554.

[22] H.T. Nguyen, M.T. Thai, T.N. Dinh, Stop-and-stare: Optimal sampling algorithms for viral marketing in billion-scale networks, in: Proceedings of the 2016 International Conference on Management of Data, SIGMOD Conference 2016, San Francisco, CA, USA, June 26 - July 01, 2016, 2016, pp. 695–710.

[23] J. Tang, X. Tang, X. Xiao, J. Yuan, Online processing algorithms for influence maximization, in: Proceedings of the 2018 International Conference on Management of Data, SIGMOD '18, Association for Computing Machinery, New York, NY, USA, 2018, pp. 991–1005.

[24] N. Chen, On the approximability of influence in social networks, SIAM J. Discr. Math. 23 (3) (2009) 1400–1415.

[25] A. Goyal, F. Bonchi, L.V.S. Lakshmanan, S. Venkatasubramanian, On minimizing budget and time in influence propagation over social networks, Soc. Netw. Anal. Min. 3 (2) (2013) 179–192.

[26] X. Wang, Y. Zhang, W. Zhang, X. Lin, Efficient distance-aware influence maximization in geo-social networks, IEEE Trans. Knowl. Data Eng. 29 (3) (2017) 599–612.

[27] M. Zhong, Q. Zeng, Y. Zhu, J. Li, T. Qian, Sample location selection for efficient distance-aware influence maximization in geo-social networks, in: Database Systems for Advanced Applications - Proceedings of the 23rd International Conference, DASFAA 2018, Gold Coast, QLD, Australia, May 21-24, 2018, Part I, 2018, pp. 355–371.

[28] W. Chen, L.V.S. Lakshmanan, C. Castillo, Information and Influence Propagation in Social Networks, in: Synthesis Lectures on Data Management, Morgan & Claypool Publishers, 2013.
[29] W. Chen, W. Lu, N. Zhang, Time-critical influence maximization in social networks with time-delayed diffusion process, in: Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, July 22-26, 2012, Toronto, Ontario, Canada, 2012.
[30] A. Borodin, Y. Filmus, J. Oren, Threshold models for competitive influence in social networks, in: Internet and Network Economics - Proceedings of the 6th International Workshop, WINE 2010, Stanford, CA, USA, December 13-17, 2010. Proceedings, 2010, pp. 539–550.
[31] A. Goyal, W. Lu, L.V.S. Lakshmanan, SIMPATH: an efficient algorithm for influence maximization under the linear threshold model, in: D.J. Cook, J. Pei, W. Wang, O.R. Zaïane, X. Wu (Eds.), Proceedings of the 11th IEEE International Conference on Data Mining, ICDM 2011, Vancouver, BC, Canada, December 11-14, 2011, IEEE Computer Society, 2011, pp. 211–220.
[32] X. Zhang, J. Zhu, Q. Wang, H. Zhao, Identifying influential nodes in complex networks with community structure, Knowl.-Based Syst. 42 (2013) 74–84.
[33] A. Bouyer, H.A. Beni, B. Arasteh, Z. Aghaee, R. Ghanbarzadeh, FIP: a fast overlapping community-based influence maximization algorithm using probability coefficient of global diffusion in social networks, Expert Syst. Appl. 213 (Part) (2023) 118869.
[34] M. Gong, J. Yan, B. Shen, L. Ma, Q. Cai, Influence maximization in social networks based on discrete particle swarm optimization, Inform. Sci. 367–368 (2016) 600–614.
[35] K. Zhang, H. Du, M.W. Feldman, Maximizing influence in a social network: Improved results using a genetic algorithm, Physica A 478 (2017) 20–30.
[36] D. Bucur, G. Iacca, Influence maximization in social networks with genetic algorithms, in: G. Squillero, P. Burelli (Eds.), Applications of Evolutionary Computation - Proceedings of the 19th European Conference, EvoApplications 2016, Porto, Portugal, March 30 - April 1, 2016, Proceedings, Part I, in: Lecture Notes in Computer Science, vol. 9597, Springer, 2016, pp. 379–392.
[37] C. Borgs, M. Brautbar, J.T. Chayes, B. Lucier, Maximizing social influence in nearly optimal time, in: Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014, 2014, pp. 946–957.
[38] H. Nguyen, R. Zheng, On budgeted influence maximization in social networks, IEEE J. Sel. Areas Commun. 31 (6) (2013) 1084–1094.
[39] N. Barbieri, F. Bonchi, G. Manco, Topic-aware social influence propagation models, Knowl. Inf. Syst. 37 (2013) 555–584.
[40] S. Chen, J. Fan, G. Li, J. Feng, K. Tan, J. Tang, Online topic-aware influence maximization, Proc. VLDB Endow. 8 (6) (2015) 666–677.
[41] G. Li, S. Chen, J. Feng, K. lee Tan, W.-S.L. and, Efficient location-aware influence maximization, in: Proceedings of the 34th IEEE International Conference on Data Engineering, ICDE 2018, Paris, France, April 16-19, 2018, 2018, pp. 1569–1572.
[42] S. Bharathi, D. Kempe, M. Salek, Competitive influence maximization in social networks, in: Internet and Network Economics: Proceedings of the Third International Workshop, WINE 2007, San Diego, CA, USA, December 12-14, 2007. Proceedings 3, Springer, 2007, pp. 306–311.
[43] B. Liu, G. Cong, D. Xu, Y. Zeng, Time constrained influence maximization in social networks, in: Proceedings of the 12th IEEE International Conference on Data Mining, ICDM 2012, Brussels, Belgium, December 10-13, 2012, 2012, pp. 439–448.
[44] J. Zhu, S. Ghosh, W. Wu, Group influence maximization problem in social networks, IEEE Trans. Comput. Soc. Syst. 6 (6) (2019) 1156–1164.
[45] Y. Lin, W. Chen, J.C.S. Lui, Boosting information spread: An algorithmic approach, in: Proceedings of the 33rd IEEE International Conference on Data Engineering, ICDE 2017, San Diego, CA, USA, April 19-22, 2017, IEEE Computer Society, 2017, pp. 883–894.
[46] Z. Wang, Y. Yang, J. Pei, L. Chu, E. Chen, Activity maximization by effective information diffusion in social networks, IEEE Trans. Knowl. Data Eng. 29 (11) (2017) 2374–2387.
[47] L.N. Nguyen, K. Zhou, M.T. Thai, Influence maximization at community level: A new challenge with non-submodularity, in: Proceedings of the 39th IEEE International Conference on Distributed Computing Systems, ICDCS 2019, Dallas, TX, USA, July 7-10, 2019, 2019, pp. 327–337.
[48] W. Lu, W. Chen, L.V.S. Lakshmanan, From competition to complementarity: Comparative influence diffusion and maximization, 2015, CoRR abs/1507.00317.
[49] A. Das, D. Kempe, Submodular meets spectral: Greedy algorithms for subset selection, sparse approximation and dictionary selection, in: L. Getoor, T. Scheffer (Eds.), Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011, Omni Press, 2011, pp. 1057–1064.
[50] A.A. Bian, J.M. Buhmann, A. Krause, S. Tschiatschek, Guarantees for greedy maximization of non-submodular functions with applications, in: Proceedings of the 34th International Conference on Machine Learning - Volume 70, ICML '17, JMLR.org, 2017, pp. 498–507.
[51] B. Lehmann, D. Lehmann, N. Nisan, Combinatorial auctions with decreasing marginal utilities, Games Econom. Behav. 55 (2) (2006) 270–296.
[52] Y. Wang, D. Xu, Y. Wang, D. Zhang, Non-submodular maximization on massive data streams, J. Global Optim. 76 (4) (2020) 729–743.
[53] U. Feige, A threshold of ln n for approximating set cover, J. ACM 45 (4) (1998) 634–652.
[54] F.R.K. Chung, L. Lu, Survey: Concentration inequalities and martingale inequalities: A survey, Internet Math. 3 (1) (2006) 79–127.
[55] J. Leskovec, Krevl, A. SNAP datasets: Stanford large network dataset collection, 2014.
[56] K. Han, Y. He, K. Huang, X. Xiao, S. Tang, J. Xu, L. Huang, Best bang for the buck: Cost-effective seed selection for online social networks, IEEE Trans. Knowl. Data Eng. (2019) 1.
[57] X. Li, J.D. Smith, T.N. Dinh, M.T. Thai, TipTop: (almost) exact solutions for influence maximization in billion-scale networks, IEEE/ACM Trans. Netw. 27 (2) (2019) 649–661.
[58] P. Dagum, R.M. Karp, M. Luby, S.M. Ross, An optimal algorithm for Monte Carlo estimation, SIAM J. Comput. 29 (5) (2000) 1484–1496.
[59] S. Sachdeva, N.K. Vishnoi, Approximation theory and the design of fast algorithms, 2013, CoRR abs/1309.4882. arXiv:1309.4882.