Groups Influence with Minimum Cost in Social Networks

Phuong N. H. Pham[1], Canh V. Pham[2(✉)], Hieu V. Duong[2], Trung Thanh Nguyen[2], and My T. Thai[3]

[1] Faculty of Information Technology, Ho Chi Minh city University of Food Industry, Ho Chi Minh, Vietnam
phuongpnh@hufi.edu.vn

[2] ORlab, Faculty of Computer Science, Phenikaa University, Hanoi 12116, Vietnam
{canh.phamvan,thanh.nguyentrung}@phenikaa-uni.edu.vn, dvhieubg95@gmail.com

[3] Department of Computer and Information Science and Engineering, University of Florida, Gainesville, USA
mythai@cise.ufl.edu

Abstract. This paper studies a Group Influence with Minimum cost which aims to find a seed set with smallest cost that can influence all target groups, where each user is associated with a cost and a group is influenced if the total score of the influenced users belonging to the group is at least a certain threshold. As the group-influence function is neither submodular nor supermodular, theoretical bounds on the quality of solutions returned by the well-known greedy approach may not be guaranteed. To address this challenge, we propose a bi-criteria polynomial-time approximation algorithm with high certainty. At the heart of the algorithm is a novel group reachable reverse sample concept, which helps speed up the estimation of the group influence function. Finally, extensive experiments conducted on real social networks show that our proposed algorithm outperform the state-of-the-art algorithms in terms of the objective value and the running time.

Keywords: Viral marketing · Group influence · Approximation algorithm · Online social network

1 Introduction

Information diffusion in Online Social Networks (OSNs) is a central research topic due to its tremendous commercial value. By leveraging the "word of mouth" effect, companies and organizations have used social networks as an effective mean of communication to promote products, spread opinion and renovation, persuade voters, etc. In a seminal work [10] published almost twenty years ago, Kempe *et al.* introduced the Influence Maximization (IM) problem, which aims to find a set of k users (called *seed set*) in a social network to initiate a propagation process that can influence a largest possible number of users, under some

D. Mohaisen and R. Jin (Eds.): CSoNet 2021, LNCS 13116, pp. 231–242, 2021.
https://doi.org/10.1007/978-3-030-91434-9_21

predefined propagation model. Since then, this problem and its notable variants have demonstrated theirs significant role in various real-world problems, not only in viral marketing [12,15], but also in other fields such as epidemics control in social network [13,20], social network monitoring [25], recommendation system [24], etc. As in many realistic scenarios, users decision and behavior tends to be dependent on his/her group and most of important decisions or works, which would affect to many individuals, are done by a group of key persons. Therefore, creating an impact on groups or communities would be able to bring more benefits than individuals, and deserves a special consideration.

Motivated by aforementioned phenomenon, recent studies have been carried out on a general version of IM, whose objective is to maximize the number of influenced groups of users instead of the number of influenced individuals, by choosing some seed set of at most k users (see, e.g. [7,17,23,27,28]). Also, one can consider a dual problem of this problem by asking for the minimum number of seed nodes to influence a given number of groups. Along this line, this paper investigates a slightly general problem, named *Groups Influence with Minimum cost* (GIM), which aims to find a seed set of minimum cost nodes to influence all the groups in the network. Different from existing works, we consider the role of each user in a group by assigning a score to him/her, and each group admits a *threshold* representing how difficult it is to be influenced. Specifically, a group is influenced if the total score of the influenced members reaches its threshold.

One can easily seen that GIM subsumes IM and its dual version as special cases and 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 group influence function (denoted by $\sigma(\cdot)$). Another challenge is that $\sigma(\cdot)$ is neither a submodular nor suppermodular, implying that the classical greedy algorithms when being applied to GIM may not result in any approximation guarantee.

In this work, we address the above challenges and our contributions can be summarized as follows. Assume that $G = (V, E)$ is a social network under a diffusion model, $\mathcal{C} = \{C_1, C_2, \ldots, C_K\}$ is a set of target groups, each group C_i has a threshold t_i, and each node u has a cost $c(u)$ and a score $b(u)$.

– We first show that the group influence function, denoted by $\sigma(\cdot)$, is neither submodular nor suppermodular, and then develop a novel sampling technique, named *Group Reachable Reverse* (GRR), to estimate $\sigma(\cdot)$. This technique plays an importance role in our proposed algorithm.
– We devise a bi-criteria approximation algorithm, named *Groups Influence Approximation* (GIA) by proposing an algorithmic framework for generating multiple candidate solutions with theoretical bounds. Specifically, GIA is a $(O(\ln K + \ln(n \ln n)), 1 - \epsilon)$-bicriteria approximation with high probability, that is, GIA runs in polynomial time and returns a solution S satisfying $c(S) \leq O(\ln K + \ln(n \ln n))\mathsf{OPT}$, and $\sigma(S) \geq (1 - \epsilon)K$ with high probability (w.h.p), where OPT is the total cost of an optimal solution, and ϵ is any fixed positive constant.
– We conduct extensive experiments on real social networks to demonstrate the effectiveness and scalability of our proposed algorithms. It is shown that our algorithms provide significantly higher quality solutions than existing

methods, while their running time is several times faster than that of the state-of-the art algorithms.

Organization. The rest of our paper is organized as follows. We give a short literature review in Sect. 2. In Sect. 3, we introduce the information diffusion, problem formulation and sampling method to estimate group influence function. Section 4 presents our main algorithm. The experiments are presented in Sect. 6 and the conclusion is given in Sect. 5.

2 Related Work

Kempe *et al.* [10] first introduced the Influence Maximization (IM) problem as a discrete optimization problem under two classical information diffusion models: Independent Cascade (IC) and Linear Threshold (LT). There are two main challenges: (1) IM is **NP**-hard and it cannot be approximated within a ratio of $1 - 1/e + \epsilon$ for any $\epsilon > 0$; (2) calculating influence spread of a seed node is #**P**-hard [3]. This work has inspired a vast mount of studies on developing efficient algorithms for IM [4,14,21,22] as well as its variants [9,15]. By utilizing monotonic and submodularity of the influence spread function, [10] also proposed a naive greedy algorithm providing an approximation ratio of $1 - 1/e$. Several fast heuristic algorithms were proposed for the large-scale networks [3], without any theoretical guarantees. In another work, Borgs *et al.* [2] introduced the concept of Reverse Influence Sampling (RIS) which paves the way to the development of a linear time $(1 - 1/e - \epsilon)$-approximation algorithm (w.h.p). Subsequent works have been considered for reducing the sampling complexity and running time by improving the RIS algorithm [14,21,22]. In other direction, several works have extended IM to different variants, such as, budget constraint [16], topic queries [1], competitive influence [18], and misinformation detection [19,26].

The groups influence (or community) maximization is one of IM variations, which has gained much attention recently. In this context, every user on social media usually belongs to a particular group and his/her behavior is influenced by those groups, making the creation of an influence on groups of users that reap more benefits than individuals. Nguyen *et al.* [17] aimed to find a seed set of k nodes that influences the largest number of communities and show that the group influence function is neither submodular nor subpermodular, and developed several approximation algorithms to solve this problem. The authors in [27,28] investigated the problem of Group Influence Maximization, which is to select k seed users such that the number of eventually activated groups is maximized, in which, a group becomes active if β percent of nodes in this group are influenced. Later, Tsang *et al.* [23] considered the influence maximization problem with several fairness constraints and propose an algorithmic framework which utilizes monotonic and submodular multi-objective functions techniques to give the approximate solutions. More recently, [7] proposed the mix integer programming approach for seeking the exact solution of this problem but it only applies to a specific set of sample graphs due to its high complexity. These studies focused on the problem of maximizing the influence of groups with limited budgets, which is different from our problem. Therefore, the existing algorithms cannot readily be applied to our problem.

3 Diffusion Models and Problem Definition

3.1 Independent Cascade Model

We model an OSN as a directed graph $G = (V, E)$ where V is the set of nodes and E is the set of edges with $|V| = n$ and $|E| = m$. Let $N_{in}(v)$ and $N_{out}(v)$ be the set of in-neighbors and out-neighbor of node v, respectively. Given a seed set (initial influenced nodes) S, an information diffusion process happens in the network and hence more nodes can be activated. In this paper we focus on the IC model but our approach can be modified to handle the LT model as well. In 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 a node v. The diffusion process from S happens in discrete time steps $t = 1, 2, \ldots$, as follows. At round $t = 0$, all nodes in S are *active* and other nodes in $V \backslash S$ are *inactive*. At step $t \geq 1$, for each node u activated at step $t - 1$, it 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 ends at step t if there is no new node is activated in this step.

The IC model is equivalent to a *live-edge* model defined as follows. From the graph $G = (V, E)$, we generate a random sample graph g by selecting edge $e \in E$ with probability $p(e)$ and not selecting e with probability $1 - p(e)$. We refer to g as a sample of G and write $g \sim G$. The probability that g is generated from G is $\Pr[g \sim G] = \prod_{e \in E(g)} p(e) \prod_{e \notin E(g)} (1 - p(e))$, 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 $\mathbb{I}(S, u) = \sum_{g \sim G} \Pr[g \sim G] \cdot r(S, u)$, where $r(S, u) = 1$ if u is reachable from S in g and $r(S, u) = 0$ otherwise. The influence spread of S in network G is: $\mathbb{I}(S) = \sum_{u \in V} \mathbb{I}(S, u)$.

3.2 Problem Definition

We are given a social network $G = (V, E)$ under the IC model and a collection of K 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 of nodes (i, j) with $i \neq j$. Denote by $C(u)$ the group that contains node u. To determine a group is influenced or not, we extend the group influence model in [17] by scoring each node in the group based on the fact that each user has a different role in his/her group. Thus, each node $u \in V$ has a *cost* $c(u)$ and a *score* $b(u)$. We denote $b_{max} = \max_{v \in V} b(v)$ and $b_{min} = \min_{v \in V} b(v)$. The weight $c(u)$ measures the cost or the price of the node u that has to pay if u is chosen as a seed node. The node score $b(u) > 0$ ($b(u)$ is an integer) indicates the role of node u in group $C(u)$. Each group C_i is assigned a threshold f t_i ($t_i > 0$), which reflects the minimum total score that we must reach if we want to influence group C_i. We say that group C_i is influenced iff the total score of influenced nodes in C_i is at least t_i. We define a *cost function* $c : 2^V \rightarrow \mathbb{R}_+$ and a *group influence function* $\sigma : 2^V \rightarrow \mathbb{R}_+$ as follows. For a given seed set $S \subseteq V$, define $c(S) = \sum_{u \in S} c(u)$ is the total cost of S and $\sigma(S)$ is the (expected) number of groups in $\mathcal{C}$ are influenced by the seed set S when the diffusion process ends, that is,

$$\sigma(S) = |\{C_i : \sum_{v \in C_i} \mathbb{I}(S, v) b(v) \geq t_i, C_i \in \mathcal{C}\}| \tag{1}$$

In the special case where each group C_i has only one node, the group influence function $\sigma(\cdot)$ above 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 we call $\sigma(\cdot)$ a supermodular function. Due to our group influence process is an extended version of [17], the function $\sigma(\cdot)$ is also neither submodular nor supermodular. We now formally define the *Groups Influence with Minimal cost* (GIM) problem as follows.

Definition 1 (GIM). *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 disjoint target groups* $\{C_1, C_2, \ldots, C_K\}$*,* $C_i \cap C_j = \emptyset$*. The objective is to find a seed set* $S \subseteq V$ *of a minimum total cost that influences all the groups in* $\mathcal{C}$*.*

It is not hard to prove the inappximibility of GIM problem, states in Theorem 1, by reducing from the classical Set Cover problem [8].

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

3.3 An Estimator of Group Influence

In this section, we first introduce the concept of *Group Reverse Reachable (GRR)* sample, by extending existing sampling methods in [2,17].

Definition 2 (GRR sample). *Given an instance of* GIM *problem* $(G, \mathcal{C})$*, a GRR sample is generated by the following four steps:*

1. *Randomly select a group* C_i *with probability* $\frac{1}{K}$ *(call* C_i *a source group).*
2. *Generate a sample graph* g *according to the live-edge model under* IC *model.*
3. *For each node* $u \in C_i$*, return a node set* $R_g(u)$ *that is reachable from* u *in* g*. We call* u *is a source node.*
4. *Return a GRR sample* $R_g = \{R_g(u) | \forall u \in C_i\}$*, we also refer to* $C(R_g)$ *as the source group of* R_g *and* $t(R_g)$ *as the threshold of* $C(R_g)$*.*

Our GRR sample is a nature extended version of the RR sample [10] by combining RR samples with the source node belongs to the source group. Moreover, our GRR sample is also an extended version of Reverse Influence Community (RIC) [17] which uses to estimate the group influence with the sore of each node is equal to 1. The main differences between ours and RIC are: (1) the definition of GRR sample specifically determines whether or not a group is influenced via the total score of influenced nodes but this is not well defined in the RIC even when the score is equal to 1, (2) storing the reachable influence set for each node in GRR sample can help us exploit some important properties that used for analyzing approximation ratio of proposed algorithms in the next sections.

For a set $S \subseteq V$ and a GRR sample R_g, and for $R_g(u) \in R_g$, if $R_g(u) \cap S \neq \emptyset$, we say that S *covers* node u, define:

$$\mathsf{cover\text{-}score}(S, R_g(u)) = b(u) \cdot \min\{|S \cap R_g(u)|, 1\} \quad (2)$$

is *score* of source node u *covered* by S in R_g, we denote following random variable:

$$X_g(S) = \begin{cases} 1, \text{ if } \sum_{R_g(u) \in R_g} \mathsf{cover\text{-}score}(S, R_g) \geq t_i \\ 0, \text{ otherwise} \end{cases} \quad (3)$$

The variable $X_g(S)$ indicates that the total score of nodes that are covered by S is greater than threshold t_i or not? When $X_g(S) = 1$, C_i is influenced by S in sample graph g. We also say that a sample R_g is influenced by S. The probability of generating a sample R_g is: $\Pr[R_g] = \frac{1}{K} \sum_{g \sim G: Re(u,g) = R_g(u), \forall u \in C(R_g)} \Pr[g \sim G]$, where $Re(u, g)$ is the set of nodes that can reach to u in g. We now show that we can estimate the value of $\sigma(S)$ by the expectation of $X_g(S)$, is a key property of GRR sample that helps us devise the algorithms.

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.*

Due to the space limitation, we omit some lemmas and proofs. From Lemma 1, we have an estimation of group influence function over a collection of GRR sets $\mathcal{R}$ is: $\hat{\sigma}(S) = \frac{K}{|\mathcal{R}|} \cdot \sum_{R_g \in \mathcal{R}} X_g(S)$.

4 Proposed Algorithm

In this section, we introduce our proposed algorithms for the GIM problem. From the analysis in Sect. 3, we can use $\hat{\sigma}(S)$ to closely estimate $\sigma(S)$ if the number of samples $|\mathcal{R}|$ is sufficiently large. Therefore, instead of solving GIM directly, we find the solution of the following problem:

Definition 3 (Samples Influence with Minimal Cost (SIM) problem). *Given a set of GRR samples $\mathcal{R}$. The problem asks to find a seed set $S \subseteq V$ with minimal total cost so that $\hat{\sigma}(S) = K$, i.e., find $S = \arg\min_{S' \subseteq V: \hat{\sigma}(S) = K} c(S')$.*

The idea behind of our algorithms is that we propose a algorithm for solving the SIM problem and use them as a core in our framework, which creates multiple candidate solutions and select a final solution. We prove the approximation guarantees by utilizing martingale theory [5].

An Approximation Algorithm for SIM. First of all, it is not hard to see that SIM problem also is NP-hard and $\hat{\sigma}_{\mathcal{R}}(\cdot)$ is non submodular and suppermodular. Therefore, similar to GIM, it does not admit a naive greedy algorithm with any approximation ratio. We handle the challenges by introducing a lower bounded function F of $\hat{\sigma}_{\mathcal{R}}(\cdot)$ and exploit its properties to devise an approximation algorithm. Define $f(S, R_g)$ is the total score of all source nodes in R_g which are influenced by set S in sample graph g: $f(S, R_g) = \sum_{u \in C(R_g)} \textsf{cover-score}(S, R_g(u))$. We can see that $f(S, R_g)$ is a non negative and monotonic set function respect to $S \subseteq V$. Define $g(S, R_g) = \min\{1, f(S, R_g)/t(R_g)\}$, we have following Lemma.

Lemma 2. *For all $S \subseteq T \subseteq V$ and $v \notin T$, $\Delta_v g(S, R_g) \geq \frac{b_{min}}{b_{max}} \cdot \Delta_v g(T, R_g)$*

Denote $\Delta_T f(S, R_g) = f(S \cup T, R_g) - f(S, R_g)$. In order to influence all samples in $\mathcal{R}$, we need to find S such that $g(S, R_g) = 1, \forall R_g \in \mathcal{R}$. Therefore, we find S with a minimal total cost such that $F(S, \mathcal{R}) = \frac{K}{T} \sum_{R_g \in \mathcal{R}} g(S, R_g) = \hat{\sigma}(S) = K$. Since $F(S, \mathcal{R})$ is a linear combination of $g(S, R_g)$, it is easy to show that $\Delta_v F(S, \mathcal{R}) \geq \frac{b_{min}}{b_{max}} \cdot \Delta_v F(T, \mathcal{R})$, for all $S \subseteq T \subseteq V$ and $v \notin V \backslash T$. $F(\cdot)$ is a lower bounded function of $\hat{\sigma}_{\mathcal{R}}(\cdot)$ and they have the same value at set S which can influence all samples in $\mathcal{R}$. We propose Modified Greedy (MoGreedy) algorithm

Algorithm 1: MoGreedy$(\mathcal{R}, \mathcal{C})$

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

1. $S \leftarrow \emptyset$
2. **while** $F(S, \mathcal{R}) < K$ **do**
3. $\quad v_{max} \leftarrow \arg\max_{v \in V \setminus S}(\min\{K, F(S \cup \{v\}, \mathcal{R})\} - F(S, \mathcal{R}))/c(v)$,
   $\quad S \leftarrow S \cup \{v_{max}\}$
4. **return** S

which utilizes the above characteristic of $F(\cdot, \mathcal{R})$. The pseudocode is presented in Algorithm 1. The general idea is that: we iteratively add a node v into the current solution S, which maximizes the marginal gain per its cost $\Delta_v(S)/c(v) = \min\{K, F(S \cup \{v\}, \mathcal{R})\} - F(S, \mathcal{R})/c(v)$ until the value of $F(S)$ achieves K.

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

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

Main Proposed Algorithm. We now present Groups Influence Approximation (GIA) algorithm, a $(1 - \epsilon, O(\ln K + \ln \ln n))$-bi criteria approximation algorithm w.h.p for GIM problem. Our algorithm is inspired by the idea of Stop-and-Stare framework for IM problem [14], which devises a stopping condition to check the quality of candidate solutions. Due to the different between GIM and IM, we introduce another stopping condition to check the candidate solutions and establish the number of required samples that ensure the theoretical bounds of the final solution. GIA algorithm operates in multiple iterations and finds

Algorithm 2: GIA algorithm

Input: Graph $G = (V, E)$, groups $\mathcal{C} = \{C_1, C_2, \dots, C_K\}$, $\epsilon, \delta \in (0, 1)$.

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

a candidate solution at each iteration by leveraging MoGreedy algorithm and checks the quality of these solutions based on static evidences. 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 operates in at most $i_{max} = \lceil\log_2(N_{max}/N_1)\rceil$ iterations, where $N_1 = (2 + \frac{2}{3}\epsilon)\frac{1}{\epsilon^2}\ln(\frac{n}{\delta})$. We then show that N_{max} is the number of samples required that can ensure the approximation ratio by Theorem 3. At iteration i, the algorithm generates 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 utilizing MoGreedy algorithm (line 5). We devise an stopping condition and check the quality of S_i in line 7. Note that, we do not reuse the stopping condition in [14], which is used in a recent work [17]. Our stopping condition is based on a lower bound of function $F_l(S, \mathcal{R}, \epsilon, \delta)$ of f, defined in Lemma 3. We show that F_l gives a lower bound value of f w.h.p in Lemma 3. The algorithm then checks termination condition in line 6. This condition also helps us prove the approximation ratio more succinctly than Stop and Stare. If the condition is true, the algorithm returns S_i as a final solution. Otherwise, it doubles size of $\mathcal{R}_i$ and moves to the next iteration. The details of the algorithm described in Algorithm 2.

Theoretical Analysis. We now analyze the performance of our algorithm.

Lemma 3. *Given* $\epsilon, \delta \in (0, 1)$*, for any set* $S \subseteq V$ *and a set of samples* $\mathcal{R}$*, denote* $c = \ln(1/\delta)$*,* $T = |\mathcal{R}|$ *we have* $\Pr[\sigma(S) \geq F_l(S, \mathcal{R}, \epsilon, \delta)] \geq 1 - \delta$*, where* $F_l(S, \mathcal{R}, \epsilon, \delta) = \min\{\hat{\sigma}_{\mathcal{R}}(S) - \frac{Kc}{3T}, \hat{\sigma}_{\mathcal{R}}(S) + \frac{K}{T}(\frac{2c}{3} - \sqrt{\frac{4c^2}{9} + 2Tc\frac{\hat{\sigma}_{\mathcal{R}}(S)}{K}})\}$.

Lemma 4. *For any set of GRR samples $\mathcal{R}$, we have:* $\hat{\sigma}_{\mathcal{R}}(S^*) = K$.

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

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* $\rho = |\bigcup_i C_i|$ *and* η *is the expectation of influence spread of a node.*

5 Experiments

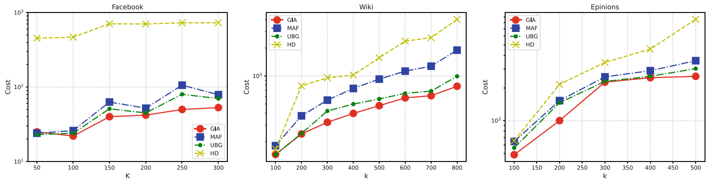
In this section, we conduct some experiments illustrating the performance of our GIA algorithm as compared with the current state-of-the-art algorithms on three metrics: the total cost of seed nodes, the time efficiency, and the value of group influence function on various network datasets.

5.1 Experimental Settings

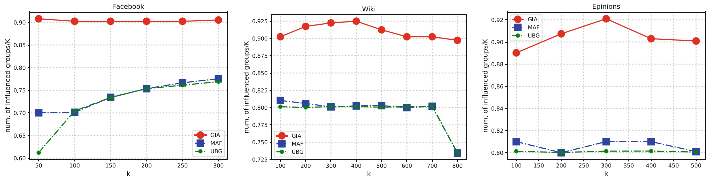
Dataset. We use three networks in recent work [17]: Facebook with 747 nodes and 60.05K edges, Wiki with 7.1K nodes and 103.6K edges and Epinions with 76K nodes and 508.8K edges.

Algorithms Compared. To our knowledge, there is no existing algorithm 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: Influence Maximization at Community level (IMC) [17]. Also, we adapt High Degree, a common baseline algorithm for related problem on information diffusion [3,4,10]. These algorithms are described in detail as follows. **UBG (Upper Bound Greedy)** [17]: This the best performance algorithm for the Influence Maximization at Community level (IMC) problem, which finds a set seed of k nodes that can influence largest the number of groups while GIM problem requires to find the set of nodes with minimal cost that can influence all target groups. Therefore, we adapt UBG algorithm 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 then add it into S until the objective is at least K. **MAF** [17]: This is also an algorithm for the IMC problem. We also modify this algorithm as for UBG to adapt for the GIM problem. **High Degree (HD):** We repeatedly select a node with highest degree until the current solution influences all target groups. For all the above algorithms, we use the Monte-Carlo method in [6] to obtain an (ϵ, δ)-approximation for estimating influence group function. For each algorithm, we run 10 times to get the average results.

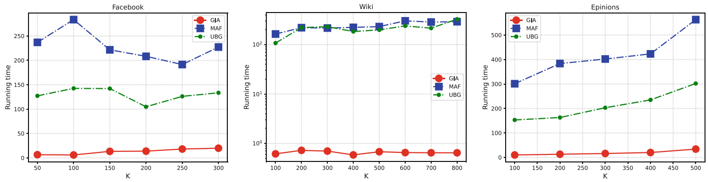
Parameters Setting. All experiments are under the IC model with edge probabilities set to $p(u,v) = 1/|N_{in}(v)|$ as in prior works [10,14,17]. We set parameters $\epsilon = 0.1$ and $\delta = 1/n$ as the default setting. For the purpose of comparing



(a) Total cost of solutions returned by algorithms



(b) The rate of expectation of influenced groups over K



(c) Running time of algorithms

Fig. 1. Performance of algorithms

our algorithm with current algorithms for the group influence problem, we set $s(u) = 1, \forall u \in U$, and the thresholds $t_i = \sum_{u \in C_i} s(u)/2$ for $i = 1 \dots K$ according to the setting in [17]. Each node has its cost calculated under Normalized Linear model with the support $(0, 1]$ according to recent works [11,16].

5.2 Experiment Results

We first compare the quality of algorithms, measured by the total cost of seed set (Fig. 1(a)). In general, GIA always provides the best solutions and outperforms other algorithms by a considerable gap. The total cost of solutions obtained by our algorithm is up to 2.12 and 1.6 times lower than those of MAF and UBG, respectively. We further report the ratio of number of influenced groups over K of algorithms in Fig. 1(b). It can be seen that GIA can output solutions with the group influence that is above $(1-\epsilon)K$ and outperforms MAF and UBG. These results show that the proposed algorithm is more efficient than the other algorithms. They do not only select the smaller-cost set of nodes but also ensure that the number of influenced group is above $(1-\epsilon)K$. Figure 1(c) shows the running time of algorithms. We do not report the running time of HD because it a simple heuristic algorithm and can finish within one second. GIA is several times faster than MAF and UBG. This is because the mechanisms of MAF and UBG consist of many iterations to find the seed set that can reach to the terminal condition. In contrast, GIA follows the mechanism of our framework which can finds the final solution after a few loops.

6 Conclusion

In this paper, we studied the GIM problem, arising from the goal of reaping the benefits of influencing user groups on social networks under a more realistic scenario. Solving the problem is challenging because of its hardness and inapproxibility and the properties of group influence function. We developed a bi-criteria approximation algorithm, called GIA, to solve GIM. The experiment results demonstrate that our algorithm outperforms the state-of-the-art ones both on the cost and on the running time.

Acknowledgements. This paper is partially supported by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under Grant No. 102.01-2020.21.

References

1. Barbieri, N., Bonchi, F., Manco, G.: Topic-aware social influence propagation models. Knowl. Inf. Syst. **37**(3), 555–584 (2013)
2. Borodin, A., Filmus, Y., Oren, J.: Threshold models for competitive influence in social networks. In: Saberi, A. (ed.) WINE 2010. LNCS, vol. 6484, pp. 539–550. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-17572-5_48
3. Chen, W., Lakshmanan, L.V.S., Castillo, C.: Information and Influence Propagation in Social Networks. Synthesis Lectures on Data Management, Morgan & Claypool Publishers, San Rafael (2013)
4. Chen, W., Yuan, Y., Zhang, L.: Scalable influence maximization in social networks under the linear threshold model. In: ICDM 2010, The 10th IEEE International Conference on Data Mining, pp. 88–97 (2010)
5. Chung, F.R.K., Lu, L.: Survey: concentration inequalities and martingale inequalities: a survey. Internet Math. **3**(1), 79–127 (2006)
6. Dagum, P., Karp, R.M., Luby, M., Ross, S.M.: An optimal algorithm for Monte Carlo estimation. SIAM J. Comput. **29**(5), 1484–1496 (2000)
7. Farnadi, G., Babaki, B., Gendreau, M.: A unifying framework for fairness-aware influence maximization. In: Companion of The 2020 Web Conference 2020, Taipei, Taiwan, 20–24 April 2020, pp. 714–722 (2020)
8. Feige, U.: A threshold of ln n for approximating set cover. J. ACM **45**(4), 634–652 (1998)
9. Goyal, A., Bonchi, F., Lakshmanan, L.V.S., Venkatasubramanian, S.: On minimizing budget and time in influence propagation over social networks. Soc. Netw. Anal. Min. **3**(2), 179–192 (2013)
10. Kempe, D., Kleinberg, J.M., Tardos, É.: Maximizing the spread of influence through a social network. In: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 137–146 (2003)
11. Li, X., Smith, J.D., Dinh, T.N., Thai, M.T.: TipTop: (almost) exact solutions for influence maximization in billion-scale networks. IEEE/ACM Trans. Netw. **27**(2), 649–661 (2019)

12. Li, Y., Zhang, D., Tan, K.: Targeted influence maximization for online advertisements. PVLDB **8**(10), 1070–1081 (2015)
13. Nguyen, H.T., Cano, A., Vu, T., Dinh, T.N.: Blocking self-avoiding walks stops cyber-epidemics: a scalable GPU-based approach. IEEE Trans. Knowl. Data Eng. **32**(7), 1263–1275 (2020)
14. Nguyen, H.T., Thai, M.T., Dinh, T.N.: Stop-and-stare: optimal sampling algorithms for viral marketing in billion-scale networks. In: International Conference on Management of Data, SIGMOD, pp. 695–710 (2016)
15. Nguyen, H.T., Thai, M.T., Dinh, T.N.: A billion-scale approximation algorithm for maximizing benefit in viral marketing. IEEE/ACM Trans. Netw. **25**(4), 2419–2429 (2017)
16. Nguyen, H., Zheng, R.: On budgeted influence maximization in social networks. IEEE J. Sel. Areas Commun. **31**(6), 1084–1094 (2013)
17. Nguyen, L.N., Zhou, K., Thai, M.T.: Influence maximization at community level: a new challenge with non-submodularity. In: 39th IEEE International Conference on Distributed Computing Systems, ICDCS 2019, pp. 327–337 (2019)
18. Pham, C.V., Duong, H.V., Bui, B.Q., Thai, M.T.: Budgeted competitive influence maximization on online social networks. In: Chen, X., Sen, A., Li, W.W., Thai, M.T. (eds.) CSoNet 2018. LNCS, vol. 11280, pp. 13–24. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-04648-4_2
19. Pham, C.V., Pham, D.V., Bui, B.Q., Nguyen, A.V.: Minimum budget for misinformation detection in online social networks with provable guarantees. Optim. Lett. 1–30 (2021)
20. Pham, C.V., Phu, Q.V., Hoang, H.X., Pei, J., Thai, M.T.: Minimum budget for misinformation blocking in online social networks. J. Comb. Optim. **38**(4), 1101–1127 (2019)
21. Tang, J., Tang, X., Xiao, X., Yuan, J.: Online processing algorithms for influence maximization. In: Proceedings of the 2018 International Conference on Management of Data, pp. 991–1005. Association for Computing Machinery (2018)
22. Tang, Y., Shi, Y., Xiao, X.: Influence maximization in near-linear time: a martingale approach. In: Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, pp. 1539–1554 (2015)
23. Tsang, A., Wilder, B., Rice, E., Tambe, M., Zick, Y.: Group-fairness in influence maximization. In: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, pp. 5997–6005 (2019)
24. Ye, M., Liu, X., Lee, W.: Exploring social influence for recommendation: a generative model approach. In: International ACM Conference on research and development in Information Retrieval, SIGIR 2012, pp. 671–680 (2012)
25. Zhang, H., Alim, M.A., Li, X., Thai, M.T., Nguyen, H.T.: Misinformation in online social networks: detect them all with a limited budget. ACM Trans. Inf. Syst. **34**(3), 18:1–18:24 (2016)
26. Zhang, H., Zhang, H., Li, X., Thai, M.T.: Limiting the spread of misinformation while effectively raising awareness in social networks. In: Thai, M.T., Nguyen, N.P., Shen, H. (eds.) CSoNet 2015. LNCS, vol. 9197, pp. 35–47. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-21786-4_4
27. Zhu, J., Ghosh, S., Wu, W.: Group influence maximization problem in social networks. IEEE Trans. Comput. Soc. Syst. **6**(6), 1156–1164 (2019)
28. Zhu, J., Ghosh, S., Wu, W., Gao, C.: Profit maximization under group influence model in social networks. In: Tagarelli, A., Tong, H. (eds.) CSoNet 2019. LNCS, vol. 11917, pp. 108–119. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-34980-6_13