TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 06/09 pp832–842
DOI: 10.26599/TST.2021.9010062
Volume 27, Number 5, October 2022

Efficient Algorithms for Maximizing Group Influence in Social Networks

Peihuang Huang, Longkun Guo*, and Yuting Zhong

Abstract: In social network applications, individual opinion is often influenced by groups, and most decisions usually reflect the majority's opinions. This imposes the group influence maximization (GIM) problem that selects k initial nodes, where each node belongs to multiple groups for a given social network and each group has a weight, to maximize the weight of the eventually activated groups. The GIM problem is apparently NP-hard, given the NP-hardness of the influence maximization (IM) problem that does not consider groups. Focusing on activating groups rather than individuals, this paper proposes the complementary maximum coverage (CMC) algorithm, which greedily and iteratively removes the node with the approximate least group influence until at most k nodes remain. Although the evaluation of the current group influence against each node is only approximate, it nevertheless ensures the success of activating an approximate maximum number of groups. Moreover, we also propose the improved reverse influence sampling (IRIS) algorithm through fine-tuning of the renowned reverse influence sampling algorithm for GIM. Finally, we carry out experiments to evaluate CMC and IRIS, demonstrating that they both outperform the baseline algorithms respective of their average number of activated groups under the independent cascade (IC) model.

Key words: complementary maximum coverage (CMC); improved reverse influence sampling (IRIS); group influence maximization (GIM); independent cascade (IC) model

1 Introduction

With the significant development in Internet technology, big data, and communication technology, including mobile and pervasive computing paradigm, communication between users has become more convenient due to the continuous advancement of Internet technology[1, 2]. Consequently, social activities are growing on the internet, with an increasing number of people expressing their opinions and sharing their daily lives on social softwares, leading to the emergence of mediated social networks. Social network platforms are essential for facilitating interaction among individuals, in particular for disseminating information and ideas. Among the most commonly used social network softwares, 2.2 billion users are on Facebook, 1 billion users are on WeChat, and 340 million users are on Twitter[3]. In contrast to real life, social network users form groups based on their common interests, hobbies, or other kinds of relationships. Individuals can join multiple groups at the same time; for example, a WeChat user can simultaneously be a member of a swimming group, family group, class group, or work group. A social network group can constitute a small number of family members or the entire population of a state/ country.

• Peihuang Huang is with the College of Mathematics and Data Science, Minjiang University, Fuzhou 350108, China. E-mail: peihuang.huang@foxmail.com.

• Longkun Guo and Yuting Zhong are with the College of Mathematics and Computer Science, Fuzhou University, Fuzhou 350116, China. E-mail: lkguo@fzu.edu.cn.

∗ To whom correspondence should be addressed.

Manuscript received: 2021-06-21; revised: 2021-07-30; accepted: 2021-08-18

Besides social networks, the influence of a group's majority has been exploited for disseminating information, advertising, marketing, and controlling public opinions. In this context, an individual's decision is influenced by the group's opinions. For instance, in the USA presidential election, the presidential candidate wins if he/she gets a majority of all the electoral votes in a state. Similarly, the chance of a company buying pens of a certain brand for all of its employees increases if the brand is trusted by most of its employees. To maximize the marketing result with limited cost in the above-mentioned examples, k number of users must be selected as advertisers for activating maximum users in a group. These applications present a problem related to maximizing real-world group influence, leading to maximizing group influence in social networks.

1.1 Related work

The problem of group influence maximization (GIM) is essentially a generalization of the influence maximization (IM) problem. For addressing the GIM problem, a few approximation algorithms are reported in the literature. Because its objective function is neither submodular nor supermodular, the greedy algorithm cannot produce a non-trivial approximation ratio[4]. In contrast, the IM problem has a submodular objective function and admits efficient constant-factor approximation algorithms. Zhu et al.[5] first proposed an algorithm with a sandwich approximation framework employing the D-SSA method for selecting seed users. Subsequently, Zhu et al.[4] introduced another sandwich approximation framework using the ED-SSA method as a building block, approximating the objective function for its upper and lower bounds, and experimentally demonstrated its advantages in comparison to the group coverage maximization algorithm. Although these breakthroughs have been made for GIM, the designing of efficient algorithms has attracted considerable research interests.

For the IM problem alone without considering groups, many researchers have proposed both heuristic and approximate algorithms aiming to activate k nodes initially for maximizing the number of resulting activated nodes. Domingos and Richardson[6] were the first to investigate the IM problem. Later, Kempe et al.[7] proved that the IM problem is NP-hard in both independent cascade (IC) and linear threshold (LT) models, although the objective function is shown submodular. They first formulated the IM problem as a discrete optimization problem, and then proposed a greedy algorithm using Monte Carlo method for simulating the influence propagation process. Notably, they achieved a performance guarantee of $\left(1-\dfrac{1}{e}-\varepsilon\right)$ for any fixed $\varepsilon > 0$, thereby attaining the best possible approximation ratio in theory. However, the greedy algorithm has a high runtime. Later, researchers developed improved heuristic algorithms and many approximation algorithms having short runtime[8–13]. Although heuristic algorithms can speedily perform the calculations, they are not theoretically accurate. Among them, the improved greedy algorithms have reduced the runtime compared with the renowned greedy algorithm based on the objective function's submodularity; however, the quality of their solution is inferior to the approximate algorithms and has no theoretical guarantees. Through intensive studies on the IM problem, many efficient algorithms have been developed. For example, Borgs et al.[14] developed the RIS algorithm to reduce the calculation time of the simulation propagation process. Tang et al.[15, 16] proposed the TIM, TIM$^+$, and IMM algorithms by improving the RIS algorithm as a prototype to ensure an approximation ratio of $\left(1-\dfrac{1}{e}-\varepsilon\right)$ under the IC model. Notably, Nguyen et al.[17] were the first ones to meet the strict theoretical threshold using the SSA and D-SSA algorithms. Both these algorithms use a minimum sample set for the IM problem. For the IM problem in social networks with special structures, faster approximation algorithms exist[18]. In addition, the GIM and IM problems are essentially coverage problems, which intensively investigated[19, 20]. However, the existing algorithms for coverage problems can not apply to the IM or GIM problems.

The community discovery algorithms decompose a social network into several communities, where the nodes within the community are connected closely, whereas the nodes distributed in different communities are connected sparsely. Therefore, the influence spreads rapidly to nodes within a community. Moreover, the optimal allocation in a social network (OASNET) algorithm and community-based greedy algorithm (CGA) are the most commonly used community discovery algorithms. To solve the IM problem using the optimal dynamic allocation of resources, Cao et al.[21] proposed the OASNET algorithm. The CGA algorithm[22] combines the existing greedy algorithm with a dynamic programming method, and allocates seed nodes to each community in a near-optimal manner for maximizing the influence of the group. For detecting a hidden community structure in the network, Ji et al.[23] presented a new algorithm by selecting the k nodes with the largest community coverage as the initial seed nodes. Furthermore, much research has been devoted to the study communities for activating maximum nodes[24–30]. Nevertheless, because the dynamic model aims to activate the largest number of groups rather than the largest number of activated individuals, the existing algorithms for the IM problem cannot be easily extended to the GIM problem.

1.2 Our contribution

The main contribution of this paper can be summarized as follows:

- A heuristic algorithm called complementary maximum coverage (CMC) is proposed for solving the GIM problem, which emphasizes the maximization of seed nodes' influence over groups collectively rather than over nodes.
- An improved reverse influence sampling (IRIS) algorithm is developed for solving the GIM problem by amending the details of the renown reverse influence sampling (RIS) algorithm.
- The CMC algorithm is evaluated by comparing with other previously existing baselines through experiments under the IC model, demonstrating the superiority of the CMC algorithm in comparison to previous algorithms considering the number of activated groups.

1.3 Organization

The remainder of this paper is organized as follows. Section 2 introduces the GIM problem formally and describes a model for representing a social network. Section 3 proposes the CMC algorithm with an execution example and then introduces the IRIS algorithm. Section 4 evaluates the proposed algorithms via numerical simulations through comparison with existing baselines. Section 5 concludes the paper.

2 Preliminaries and Problem Statement

In this section, we shall first introduce the modeling of social networks and then propose the GIM problem.

2.1 Network model

In the social network modeling, we assume $G = (V, E, P, U)$ for modeling the network.

V represents the set of nodes. Node $v_i \in V$ indicates a user in a social network. Assume that there are n users in a social network, then $V = \{v_1, v_2, \ldots, v_n\}$ represents the set of users. Moreover, a node's edges joining other nodes indicate the node's influence on other nodes or the node being influenced by other nodes. Two nodes are regarded as neighbors when there is an edge between them.

E represents the set of edges. An edge indicates the possible influence between nodes. Assume that there are m edges in a social network, then we have $E = \{e_1, e_2, \ldots, e_m\}$. In the model, an edge can be either directed or undirected. More precisely, for directed edges (u, v), node u directly influences node v, whereas the reverse is not true because u and v are the source and the destination nodes, respectively. Let the outgoing edges of u and the entrance edges of u be the edges leaving and entering u, respectively. The out-degree of u indicates the number of outgoing edges of u and the in-degree of u represents the number of edges entering u.

P stores the weights of the edges, where each weight represents the probability of the corresponding edge's activation. We assume $P = \{p_1, p_2, \ldots, p_m\}$, and $\forall p_i \in [0, 1], 1 \leqslant i \leqslant m$. A higher probability indicates that the source node of the edge is more likely to influence the target node successfully.

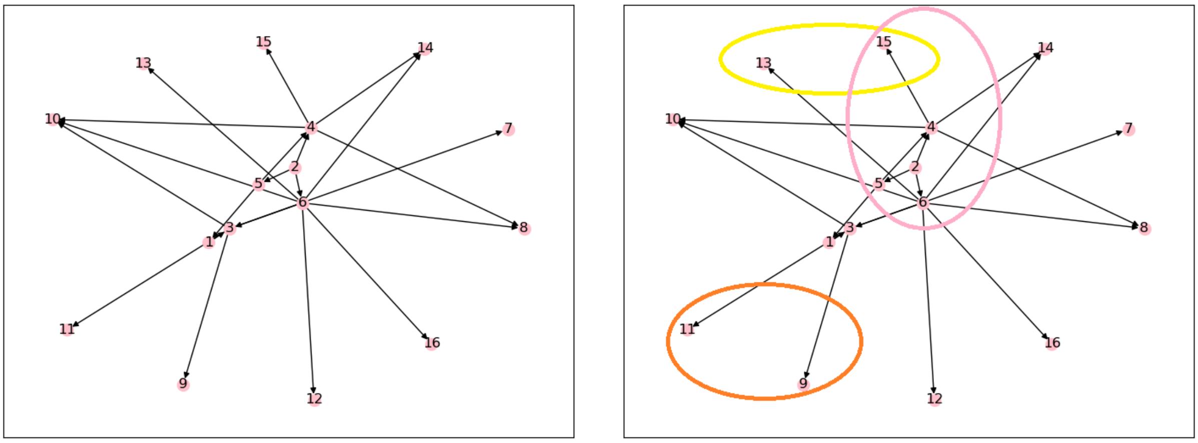
U is the set of groups. We assume that there are l groups in a social network, and denote the family of groups as $U = \{u_1, u_2, \ldots, u_l\}$, where u_j is a subset of V. Each node of the social network represents an individual that possibly belongs to one or several distinct groups. Moreover, β is assumed to be the threshold of activation, i.e. each group is successfully activated if at least β of its members are activated. The notations and definitions used in the paper are tabulated in Table 1.

2.2 Group influence maximization

As above-mentioned, the traditional IM problem (without considering groups) aims to find k initially active nodes, such that the number of eventually activated nodes would be maximized against a specified information diffusion model, say the IC model. Figure 1a illustrates a traditional social network, where each directed edge indicates the influence flows leaving the source node and entering the target node with a probability determined by the edge weight. In contrast, the GIM problem aims to activate a set of groups with maximized expected group weight, also through the k initially selected nodes. In the GIM problem, a node could appear in one or several groups, whereas a group

Table 1 Notations and definitions.

Notation	Definition
$G = (V, E, P, U)$	G is the graph modeling the social network, V is the set of nodes representing the set of users, E is the set of edges representing the influence flow, P is the set of probabilities representing probabilities of edges influence, and U is the set of groups.
p	The probability with which a node activates its neighbor, $0 \leqslant p \leqslant 1$.
$n = \|V\|$	The number of nodes
$m = \|E\|$	The number of edges
$l = \|U\|$	The number of groups
k	The bound of the number of initial seeding nodes
β	The uniform activation threshold of groups, $0\% < \beta \leqslant 100\%$
$\rho(S)$	The expected number of groups activated by seed nodes



(a) A social network without groups

(b) A social network with groups indicated by circles

Fig. 1 Illustration of social networks with and without groups.

can only be activated only if at least β of its members are activated. Note that a large β creates more difficulties in activating the group.

Clearly, activating groups is different to activating nodes. As depicted in Fig. 1b, there are three groups, say $U = \{u_1, u_2, u_3\}$, in the social network, among which u_1 is in the yellow ellipsoid, u_2 is in the pink ellipsoid, and u_3 is in the orange ellipsoid. Thus, the groups are $u_1 = \{13, 15\}$, $u_2 = \{2, 4, 5, 6, 15\}$, and $u_3 = \{9, 11\}$. It is assumed that the group activation thresholds are uniformly 50%, i.e., if at least half of a group's members are activated, it is active. Further, a seed node $\{2\}$ is assumed to activate the set of nodes $\{2, 4, 15\}$, resulting in three activated nodes. Then, according to the rule of activation of groups, u_1 and u_2 are both activated, and hence $\rho(S) = 2$. In contrast, if $\{2, 4, 5, 6\}$ are successfully influenced by the seed node $\{2\}$, then four nodes are activated. In this context, only u_2 is activated; hence $\rho(S) = 1$. Thus, activating more nodes does not necessarily result in more activated groups, although more activated nodes do likely activate more groups in most cases.

In essence, the GIM problem can be considered as a generalization of the IM problem. By setting $\beta = 100\%$ and assuming that each group in the GIM problem contains only a single node, the GIM problem immediately reduces to the IM problem. Because of the NP-hardness of the IM problem, the GIM problem is clearly NP-hard. The formal definition of the GIM problem with respect to the given graph $G = (V, E, P, U)$ can be stated as follows:

$$\begin{aligned} \max \quad & \rho(S) \\ \text{s.t.} \quad & |S| \leqslant k, \end{aligned}$$

where S is the set of candidate seed nodes, and k and $\rho(S)$ are the number of initially selected seeds and the expected weight of the consequently activated groups, respectively. Calculating $\rho(S)$ for the GIM problem is difficult because it is #P-hard to compute $\rho(S)$ under the IC model even for the IM problem.

3 Efficient Algorithms for GIM

In this section, the CMC algorithm for the GIM problem is proposed, and then the IRIS algorithm is presented.

3.1 Complementary maximum coverage algorithm

As above-mentioned, a greedy algorithm for maximum coverage that repeatedly seeks a seeding nodes (k nodes in total) to cover a maximum number of groups can be easily obtained. However, such an algorithm does not appropriately evaluate the contribution of nodes to the eventually activated groups. Consider that a node covers a maximum number of groups, where each group requires β active members to be activated. If a group is not eventually activated at the end, the node's contribution to the group should not be counted. Moreover, if a group already has more active members than the threshold β, the contribution of a node to the group should be zero.

The CMC algorithm can be considered as the complement to the above greedy algorithm. Instead of greedily adding a node with the largest contribution until all k nodes are estimated, a node with the least influence on groups is removed repeatedly. After the iterative removal of $n-k$ nodes, the k remaining nodes are obtained as seed users. The influence of a node on a group is not only reflected in whether its deletion affects the group's activation, but also in whether it can activate other members in the group. The function $f_c\ (v_i)$ denotes the influence of v_i over groups that it covers. For a node v_i belonging to none of the groups, its contribution is determined by $f_c = 0$; for a node that covers at least one group, f_c equals the sum of its contribution to the groups. If a node can provide better group coverage, then the value of f_c of the node may (but not necessarily) be large. The formulation of $f_c\ (v_i)$ for capturing the approximate influence of v_i is shown as follows:

$$f_c(v_i) = \sum_j \frac{a_{v_i}(u_j)}{|u_j| - H_{u_j} + 1} \tag{1}$$

where node v_i is in group u_j, $|u_j|$ is the number of all the members of u_j, $H_{u_j} = \beta \times |u_j|$ is the threshold for activating u_j. Moreover, $|u_j| - H_{u_j}$ implies that u_j allows $|u_j| - H_{u_j}$ nodes to be deleted, where larger $|u_j| - H_{u_j}$ indicates that v_i is less important for u_j. Since the denominator is at least one, $|u_j| - H_{u_j} + 1$ is used to define the denominator. Lastly, a_{v_i} is used to denote the number of nodes successfully activated by v_i within the remaining nodes of u_j (including v_i itself). Note that the nodes activated by v_i can be obtained using the breadth-first search (BFS) method; thus, a_{v_i} can be simply calculated as the number of such activated nodes in u_j, and consequently the result of a_{v_i} is obtained. Thus, $a_{v_i}(u_j)$ measures the contribution of v_i to the group u_j, i.e., a larger a_{v_i} possibly results in more activated nodes of group u_j, and consequently indicates an increased possibility of activating u_j.

Lemma 1 Algorithm 1 terminates in runtime $O(mnl + n^2 l)$.

Proof For the initialization, Steps 2 and 3 of the CMC algorithm compute $f_c(v_i)$ for each node v_i and construct the node-group list, where each computation traverses every node and then adds up the number of newly activated nodes for all the l groups using the BFS method. Employing an appropriate data structure, the BFS method traverses n nodes and m edges within $O(n+m)$ time, whereas $O(l)$ groups need $O(nl)$ time to calculate all the information. Hence, the initialization takes $O(n^2 l + mnl)$ time. Moreover, the while-loop will iterate for $O(n-k)$ times in the algorithm. In each iteration, the algorithm updates both $f_c(v_i)$ and the list regarding the removal of each node v_i, each of which takes $O(nl)$ time. Therefore, the CMC algorithm will terminate within a total runtime of $O(mnl + n^2 l)$. ■

3.2 An example of executing Algorithm 1

Figure 2 shows a social network containing two groups consisting of seven nodes and five directed edges with the probability of activation as 1. The instance can be modeled as a two-layer graph, where the nodes

Algorithm 1 Complementary maximum coverage algorithm

```
Require: $G = (V, E, P, U)$: an instance of GIM, $k$: the budget of seeds, and $\beta$: the activation threshold of groups.
Ensure: A set of seeds $S$.
1: Set $S := \Phi$;
2: Compute $f_c(v_i)$ for each node $(v_i)$ by Eq. (1);
3: Construct a node-group list which stores the relationship between nodes and groups, as well as the information including the current value of $f_c(v_i)$ and all $u_j$ for each $v_i$;
4: Sort the nodes according to the calculated $f_c(v_i)$;
5: While $|V| > k$ do
6:     Remove the node $(v_i)$ with minimum $f_c(v_i)$ from $V$;
7:     Update the node-group list including $f_c(v)$ for each node $v$ upon the removal of $(v_i)$;
8:     Update the order of the nodes according to updated $f_c(v)$;
9: Endwhile
10: Return $S := V$.
```

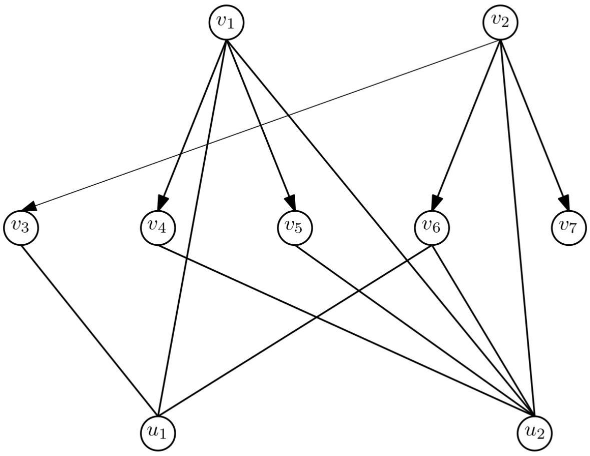


Fig. 2 A simple social network with two layers.

(users) are in the first and second upper layers, whereas the groups are in the lower layer. The upper layer is $layer_1 = \{v_1, v_2\}$, where v_1 influences $\{v_4, v_5\}$ and v_2 influences $\{v_3, v_6, v_7\}$; the lower layer is $layer_2 = \{u_1, u_2\}$, where $u_1 = \{v_1, v_3, v_6\}$ and $u_2 = \{v_1, v_2, v_4, v_5, v_6\}$. Then, $|u_1| = 3$ and $|u_2| = 5$ can be easily computed.

Assuming that $\beta = 50\%$, the contribution of all nodes is first computed using the CMC algorithm, as shown in Eq. (2) below:

$$\begin{aligned}
f_c(v_1) &= \left[\frac{0+1}{(1-50\%)\times 3+1} + \frac{2+1}{(1-50\%)\times 5+1}\right] \approx 1.08,\\
f_c(v_2) &= \frac{1+1}{(1-50\%)\times 5+1} = 0.5,\\
f_c(v_3) &= \frac{0+1}{(1-50\%)\times 3+1} \approx 0.33,\\
f_c(v_4) &= \frac{0+1}{(1-50\%)\times 5+1} = 0.25,\\
f_c(v_5) &= \frac{0+1}{(1-50\%)\times 5+1} = 0.25,\\
f_c(v_6) &= \left[\frac{0+1}{(1-50\%)\times 3+1} + \frac{0+1}{(1-50\%)\times 5+1}\right] \approx 0.58,\\
f_c(v_7) &= 0 \qquad (2)
\end{aligned}$$

Then, the computed contribution as the key attribution to sort the nodes are used, and the ordering as $\{v_1, v_6, v_2, v_3, v_4, v_5, v_7\}$ is obtained. If up to two nodes, say $k = 2$, can be used as seeds, the CMC algorithm will remove v_7 in the first iteration as it has the least contribution.

The CMC algorithm selects seed nodes based on the computed contribution of each node. The contribution depends on the covered groups and the neighbors of the nodes, e.g., executing Algorithm 1 against a social network, the removed node v_i has the least approximately computed $f_c(v_i)$, which covers the fewest groups and activates the fewest target nodes in its neighborhood.

3.3 Improved reverse influence sampling

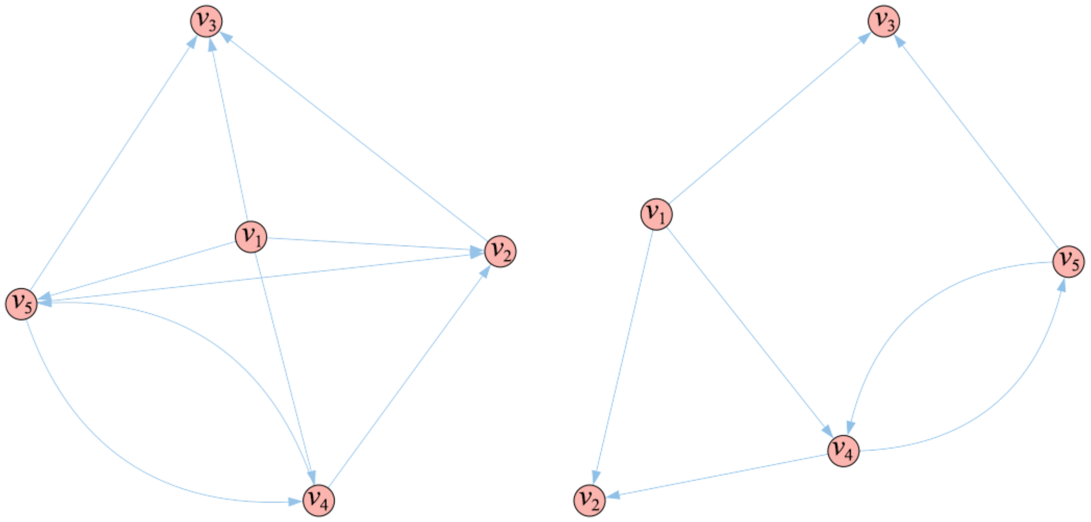
Observing that the existing RIS algorithm cannot be directly applied to solve GIM, we propose IRIS algorithm with an extension to GIM. The devised IRIS algorithm consists of two phases: the first is to generate reverse reachable (RR) sets, and the second is, indeed, to select seed nodes. In the first phase, we randomly select a node v from the graph and traverse the edges entering v. Recall that each edge has a probability of p, i.e., with probability of p, the edge is inverted, and with a probability of $1 - p$ it remains unchanged. Then, we actually generate a sparse reverse graph, which keeps the high-probability edges with a potentially wide range of propagation. Intuitively, the set of nodes, from which node v is reachable with high probability, is node v's RR set. An example is as illustrated in Fig. 3, where Fig. 3a is the original social network graph containing five nodes incident with ten directed edges, and Fig. 3b is the resultant sparse graph having only seven edges of high probability. By the algorithm, the RR set for node v_2 will be $\{v_2, v_1, v_4, v_5\}$, in which each node will have a relatively high probability of activating v_2.

Recall that the selection phase of the traditional RIS algorithm is to select the seed nodes covering maximum RR sets, since more RR sets can result in more eventually activated nodes. However, the aim of GIM is to activate a maximum weight of groups, rather than the maximum weight of nodes; but the weight of groups is not linearly dependent on the weight of nodes. Thus, in the second phase of our devised IRIS, we select k nodes, each of which covers a maximum number of groups. The formal layout of the algorithm is as illustrated in Algorithm 2.

Lemma 2 Algorithm 2 terminates within a runtime $O(\Gamma(n+m) + knl)$, where Γ is the number of random sparse graphs.

Proof For the first phase, our IRIS algorithm constructs Γ random sparse graphs of G for each of which we randomly pick a node for generating an RR set. Therefore, the first phase consumes a runtime of $O(\Gamma(n+m))$.

For the second phase, we iterate for k times where each iteration is for selecting a node with the maximized



(a) The original graph

(b) A sparse graph regarding

Fig. 3 An example of generating a reverse reachable set.

Algorithm 2 Improved reverse influence sampling

```
Require: Input of GIM including the graph $G = (V, E, P, U)$,
    the budget of seed nodes $k$, and Monte Carlo time bound $t$.
Ensure: The set $S$ of $k$ seeds.
Initialize the seed set $S := \Phi$;
Call Algorithm 3 to generate $t$ number of RR sets and add
  them to $R$;
for $i = 1$ to $k$ do
    Find in $R$ the node with maximum group coverage and
     add it into $S$;
    Delete RR sets that contain $S$ in $R$;
endfor
return $S$.
```

Algorithm 3 get_RRS algorithm

```
Require: The input of of GIM including $G = (V, E, P, U)$.
Ensure: A RR_set.
Set RR_set, $new\_nodes := \Phi$;
Construct a random graph $g$ with respect to $G$;
Uniformly and randomly select node $v$ from $g$;
Add the node $v$ into the set $new\_nodes$;
While the set $new\_nodes$ is not empty do
    Simulate influence spread, starting from $new\_nodes$;
    Add source nodes of $new\_nodes$ into RR_set;
    Update the set of $new\_nodes$;
Endwhile
Return the set of RR_set.
```

number of covered groups in the RR sets. Hence, the second step has a runtime of $O\,(knl)$. Summing up the runtime of the two phases, we obtain the total runtime $O(\Gamma(n+m)+knl)$ for IRIS. ■

4 Numerical Experiments

In this section, we evaluate our complementary maximum coverage (CMC) and improved reverse influence sampling (IRIS) algorithms under the independent cascade (IC) model, via comparison with other previously existing baselines including the maximum coverage algorithm (MC) and the maximum out-degree algorithm (MO). The algorithms were implemented with Python 3.7, and the experiments were run on a platform with Intel I5 CPU and 8G RAM memory.

4.1 Datasets

In the experiments, two datasets were used: Dataset 1 for undirected graphs and Dataset 2 for directed graphs. Dataset 1 is a collection of public information of social network users in Asia, such as Malaysia, the Philippines, Singapore, and other countries in March 2020[31]. The nodes represent users of the music streaming service LastFM, and the link between the two nodes indicates the friendship therein. In Dataset 2, there are nine snapshots of the Gnutella peer-to-peer file sharing network from SNAP in August 2002, in which the nodes and the edges represent the hosts in the Gnutella network and the connections between the hosts, respectively. In order to facilitate the experiments, we randomly generated groups as well as the probability of each edge. Observing that the IRIS algorithm is designated for directed graphs, we use Dataset 2 for all the four algorithms and Dataset 1 for other algorithms except IRIS. Table 2 displays the statistics information of the used datasets.

For Dataset 1 and Dataset 2, we observed that k and β were the main factors affecting the objective function. Therefore, we set the value of k from 5 to 80 with each increasing step being 5. Moreover, we set the values of β as 10% and 20% for Dataset 1, and set the values of β as 5%, 8%, 10%, 12%, 15%, and 18% for Dataset 2.

4.2 Experimental results

As demonstrated in Fig. 4, CMC performed slightly better than both MC and MO regarding Dataset 1, which is the dataset for undirected graphs. Observing that the goal of MO is to find k nodes with the largest out-degree and is without focusing on group activation, it is reasonable that MO performed the worst. When $\beta = 10\%$, there was only a slight difference between the performances of the CMC and MC algorithms, owing to the activation threshold being small at this point; hence both algorithms found appropriate seeding nodes to activate most groups. In summary, the CMC algorithm achieved the best performance. There is a somewhat larger difference between the experimental results of the two algorithms when $\beta = 20\%$, as it becomes more difficult to activate groups against a larger activation threshold. The enlarging gap reveals the shortcomings of the MC algorithm.

From Figs. 5a and 5b, it is demonstrated that both CMC and IRIS perform better than MC and MO, on average, for the directed graph Dataset 2. In particular, CMC has the best performance, while the experimental results of IRIS were only slightly better than those of MC. In comparison, the ranking of the four algorithms, according to the experimental results as shown in Figs. 5c and 5d, remains similar to that of Figs. 5a and 5b. Among them, CMC also significantly outperforms the other three algorithms. The gap in the experimental results between of IRIS and MC grows, while IRIS approaches those of CMC. The phenomenon suits the theoretical analysis, as IRIS focuses on finding nodes emphasizing both the weight of covered groups and the spreading influence.

Figures 5e and 5f shows that our CMC algorithm outperforms all the other algorithms, except for IRIS in some special scenarios. The occurrence of abnormal situations is mainly due to the different seed nodes obtained by IRIS in iterations. The RR sets calculated by IRIS focus on the influence of inter-group nodes, while CMC emphasizes the influence between intra-group nodes.

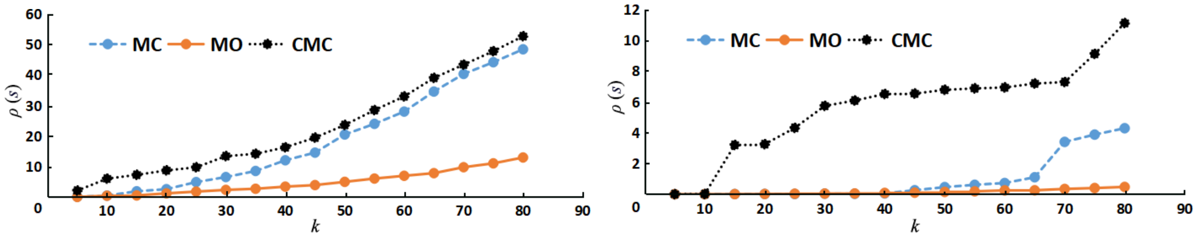
To summarize, the average performance of CMC in terms of the number of activation groups is superior to that of other algorithms regarding both Dataset 1 and Dataset 2. In all the experimental instances (particularly for Dataset 2), the performance of both CMC and IRIS is superior to that of the other algorithms. We note that CMC is outperformed by IRIS in some cases when $\beta \geqslant 15\%$, revealing that the advantage of CMC is not significant for relatively large β when compared to IRIS. As for runtime, as demonstrated in the experiments, the three algorithms CMC, MC, and MO, run faster than IRIS. Although CMC is the third fastest, it is only slightly slower than MO and MC, and, notably, it is much faster than IRIS in all the instances in the experiments. Combining the solution quality and runtime performance, we conclude that CMC is the best one that outperforms all the other three algorithms.

5 Conclusion

In the paper, we first proposed CMC, a heuristic algorithm for GIM, with the key idea of removing the least influence node over the groups until there are only

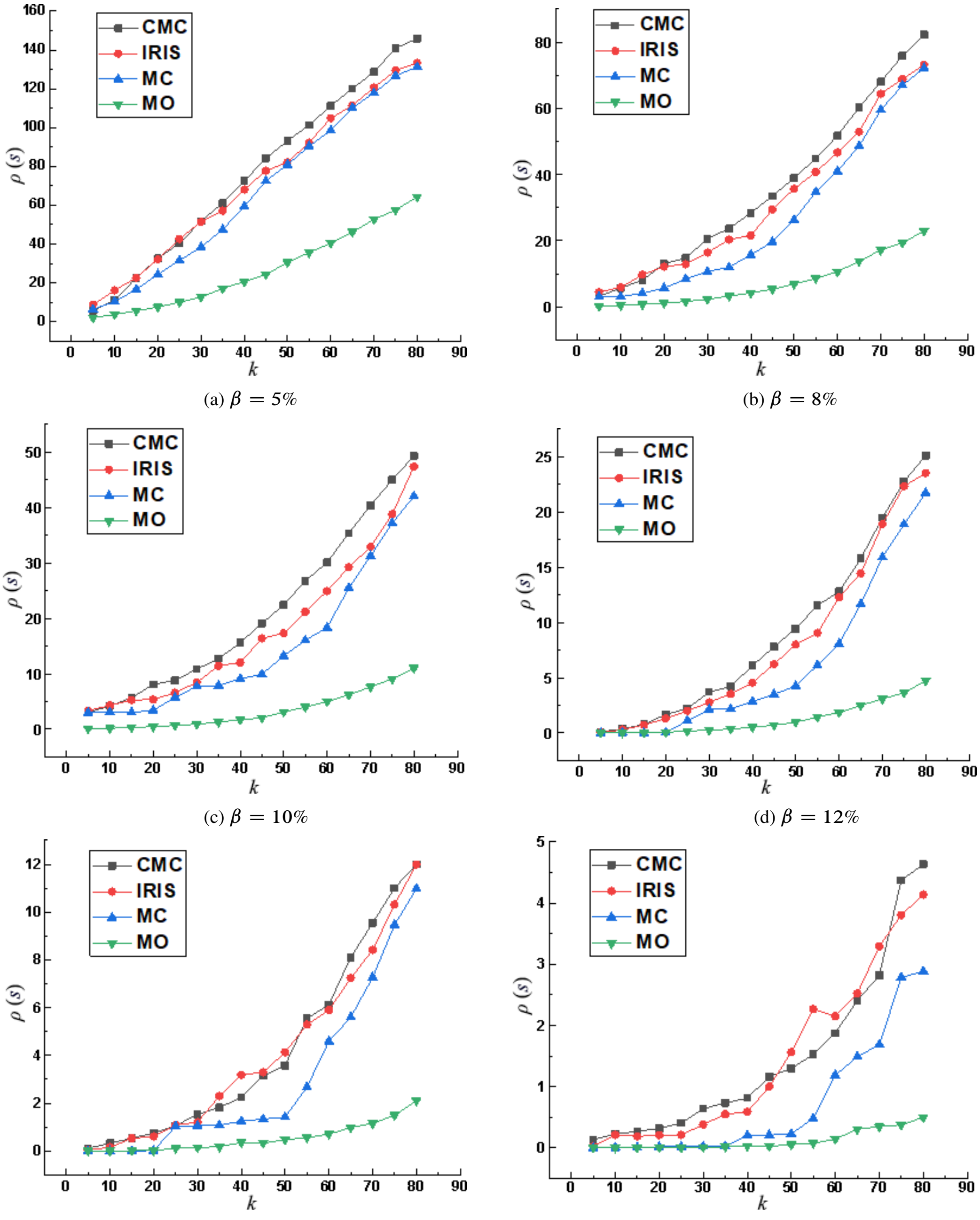
Table 2 Dataset information.

Dataset	Type	Number of nodes	Number of edges	Number of groups	Average group size
1	Undirected	7624	27806	198	34.01
2	Directed	6301	20777	234	33.84



(a) $\beta = 10\%$ (b) $\beta = 20\%$

Fig. 4 Comparison of CMC, MC, and MO under IC model for Dataset 1.



(a) $\beta = 5\%$

(b) $\beta = 8\%$

(c) $\beta = 10\%$

(d) $\beta = 12\%$

(e) $\beta = 15\%$

(f) $\beta = 18\%$

Fig. 5 Comparison of CMC, IRIS, MC, and MO under IC model for Dataset 2.

k nodes left. An algorithm called IRIS was presented in collaboration with the subroutine of analyzing the influence of each node on the groups, and ensuring activation of an approximate maximum number of groups. In essence, the IRIS algorithm was derived via improving the previously existing RIS algorithm. Finally, we evaluated the performance of CMC and IRIS, comparing them with previously existing baselines by performing experiments that demonstrated the number of their average activated groups. We are currently striving to design an approximation algorithm with a theoretical performance guarantee.

Acknowledgment

This work was supported by the Natural Science Foundation of Fujian Province (No. 2020J01845), the Educational Research Project for Young and Middle-Aged Teachers of Fujian Provincial Department of Education (No. JAT190613), the National Natural Science Foundation of China (Nos. 61772005 and 92067108), and the Outstanding Youth Innovation Team Project for Universities of Shandong Province (No. 2020KJN008).

References

[1] A. Belhassena and H. Z. Wang, Trajectory big data processing based on frequent activity, *Tsinghua Science and Technology*, vol. 24, no. 3, pp. 317–332, 2019.

[2] J. Li, M. Siddula, X. Z. Cheng, W. Cheng, Z. Tian, and Y. S. Li, Approximate data aggregation in sensor equipped IoT networks, *Tsinghua Science and Technology*, vol. 25, no. 1, pp. 44–55, 2019.

[3] M. Meeker and L. Wu, Internet trends 2018, Tehnical report, Kleiner Perkins, 2018.

[4] J. M. Zhu, S. Ghosh, and W. L. Wu, Group influence maximization problem in social networks, *IEEE Transactions on Computational Social Systems*, vol. 6, no. 6, pp. 1156–1164, 2019.

[5] J. M. Zhu, J. L. Zhu, S. Ghosh, W. L. Wu, and J. Yuan, Social influence maximization in hypergraph in social networks, *IEEE Transactions on Network Science and Engineering*, vol. 6, no. 4, pp. 801–811, 2019.

[6] P. Domingos and M. Richardson, Mining the network value of customers, in *Proc. 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, CA, USA, 2001, pp. 57–66.

[7] D. Kempe, J. Kleinberg, and E. Tardos, Maximizing the spread of influence through a social network, in *Proc. 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Washington, DC, USA, 2003, pp. 137–146.

[8] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance, Cost-effective outbreak detection in networks, in *Proc. 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Jose, CA, USA, 2007, pp. 420–429.

[9] A. Goyal, W. Lu, and L. V. S. Lakshmanan, Celf++: Optimizing the greedy algorithm for influence maximization in social networks, in *Proc. 20th International Conference Companion on World Wide Web*, Hyderabad, India, 2011, pp. 47–48.

[10] W. Chen, Y. J. Wang, and S. Y. Yang, Efficient influence maximization in social networks, in *Proc. 15th ACM SIGKDD International Conference On Knowledge Discovery and Data Mining*, Paris, France, 2009, pp. 199–208.

[11] S. Wasserman and K. Faust, Social network analysis in the social and behavioral sciences, in *Social Network Analysis: Methods and Applications*, S. Wasserman, ed. Cambridge, UK: Cambridge University Press, 1994, pp. 1–27.

[12] P. A. Estevez, P. Vera, and K. Saito, Selecting the most influential nodes in social networks, presented at 2007 International Joint Conference on Neural Networks, Orlando, FL, USA, 2007.

[13] T. H. Hubert Chan, L. Ning, and Y. Zhang, Influence maximization under the non-progressive linear threshold model, presented at 14th International Workshop on Frontiers in Algorithmics, Haikou, China, 2020.

[14] C. Borgs, M. Brautbar, J. Chayes, and B. Lucier, Maximizing social influence in nearly optimal time, in *Proc. 25th Annual ACM-SIAM Symposium on Discrete Algorithms*, Portland, OR, USA, 2014, pp. 946–957.

[15] Y. Z. Tang, X. K. Xiao, and Y. C. Shi, Influence maximization: Near-optimal time complexity meets practical efficiency, in *Proc. 2014 ACM Sigmod International Conference On Management of Data*, Sniwbird, UT, USA, 2014, pp. 75–86.

[16] Y. Z. Tang, Y. C. Shi, and X. K. Xiao, Influence maximization in near-linear time: A martingale approach, in *Proc. 2015 ACM SIGMOD International Conference on Management of Data*, Melboume, Australia, 2015, pp. 1539–1554.

[17] H. T. Nguyen, M. T. Thai, and T. N. Dinh, Stop-and-stare: Optimal sampling algorithms for viral marketing in billion-scale networks, in *Proc. 2016 International Conference on Management of Data*, San Francisco, CA, USA, 2016, pp. 695–710.

[18] J. Wu and N. Wang, Approximating special social influence maximization problems, *Tsinghua Science and Technology*, vol. 25, no. 6, pp. 703–711, 2020.

[19] Y. Zhang, Q. Ge, R. Fleischer, T. Jiang, and H. Zhu, Approximating the minimum weight weak vertex cover, *Theoretical Computer Science*, vol. 363, no. 1, pp. 99–105, 2006.

[20] Y. Zhang and H. Zhu, Approximation algorithm for weighted weak vertex cover, *Journal of Computer Science and Technology*, vol. 19, no. 6, pp. 782–786, 2004.

[21] T. Y. Cao, X. D. Wu, S. Wang, and X. H. Hu, OASNET: An optimal allocation approach to influence maximization in modular social networks, in *Proc. 2010 ACM Symposium on Applied Computing*, Sierre, Switzerland, 2010, pp. 1088–1094.

[22] Y. Wang, G. Cong, G. J. Song, and K. Q. Xie, Community-based greedy algorithm for mining top-k influential nodes in mobile social networks, in *Proc. 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington*, DC, USA, 2010, pp. 1039–1048.

[23] J. C. Ji, L. Huang, Z. Wang, H. M. Li, and S. Y. Li, A new approach to maximizing the spread of influence based on community structure, (in Chinese), *Journal of Jilin University (Science Edition)*, vol. 49, no. 1, pp. 93–97, 2011.

[24] S. Wang, B. Li, X. J. Liu, and P. Hu, Division of community-based influence maximization algorithm, (in Chinese), *Computer Engineering and Applications*, vol. 52, no. 19, pp. 42–47, 2016.

[25] J. X. Shang, S. B. Zhou, X. Li, L. C. Liu, and H. C. Wu, CoFIM: A community-based framework for influence maximization on large-scale networks, *Knowledge-Based Systems*, vol. 117, pp. 88–100, 2017.

[26] J. R. Xie, B. K. Szymanski, and X. M. Liu, SLPA: Uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process, presented at 2011 IEEE 11th International Conference on Data Mining Workshops, Vancouver, Canada, 2011, pp. 344–349.

[27] L. Q. Qiu, W. Jia, and X. Fan, Influence maximization algorithm based on overlapping community, *Data Analysis and Knowledge Discovery*, vol. 3, no. 7, pp. 94–102, 2019.

[28] Y. C. Chen, W. Y. Zhu, W. C. Peng, W. C. Lee, and S. Y. Lee, CIM: Community-based influence maximization in social networks, *ACM Transactions on Intelligent Systems and Technology*, vol. 5, no. 2, pp. 1–31, 2014.

[29] H. M. Huang, H. Shen, Z. Q. Meng, H. J. Chang, and H. W. He, Community-based influence maximization for viral marketing, *Applied Intelligence*, vol. 49, no. 6, pp. 2137–2150, 2019.

[30] A. Bozorgi, S. Samet, J. Kwisthout, and T. Wareham, Community-based influence maximization in social networks under a competitive linear threshold model, *Knowledge-Based Systems*, vol. 134, pp. 149–158, 2017.

[31] B. Rozemberczki and R. Sarkar, Characteristic functions on graphs: Birds of a feather, from statistical descriptors to parametric models, in *Proc. 29th ACM International Conference on Information & Knowledge Management*, Virtual Event Ireland, 2020, pp. 1325–1334.

Peihuang Huang received the BS degree in mathematics from Fujian Normal University in 2005, and PhD degree in intelligent systems from Fuzhou University in 2019. She is currently an associate professor in the College of Mathematics and Data Science, Minjiang University. She has published more than 20 academic papers with major research interests including sensor networks and networking science.

Yuting Zhong received the BS degree in management science and the ME degree in computer science from Fuzhou University in 2017 and 2020, respectively. She is currently a senior engineer in Ruijie Networks Co., Ltd. Her major research interests include social networks and influence maximization.

Longkun Guo received the BS and PhD degrees in computer science from University of Science and Technology of China (USTC) in 2005 and 2011, respectively. He had been a research associate of the University of Adelaide from 2015 to 2016, and is currently a full professor in the College of Mathematics and Computer Science, Fuzhou University. He has published more than 80 academic papers in reputable journals/conferences such as *Algorithmica*, IEEE TMC, IEEE TC, IEEE TPDS, IEEE ICDCS, IJCAI, and SPAA. His major research interests include efficient algorithm design and computational complexity analysis, particularly for optimization problems in high performance computing systems and networks, VLSI, etc.