RESEARCH-ARTICLE

Integrating Group Consensus for Competitive Influence Maximization in OSNs

GUOBANG CHEN, Hunan University, Changsha, Hunan, China

WENJUN JIANG, Hunan University, Changsha, Hunan, China

KENLI LI, Hunan University, Changsha, Hunan, China

JINGJING WANG, Changsha University, Changsha, Hunan, China

JIE WU, Temple University, Philadelphia, PA, United States

KIANLEE TAN, National University of Singapore, Singapore City, Singapore

Open Access Support provided by:

Hunan University

Temple University

Changsha University

National University of Singapore

Published: 06 November 2025
Online AM: 18 September 2025
Accepted: 11 September 2025
Revised: 15 July 2025
Received: 14 February 2025

Integrating Group Consensus for Competitive Influence Maximization in OSNs

GUOBANG CHEN, WENJUN JIANG, and KENLI LI, College of Computer Science and Electronic Engineering, Hunan University, Changsha, China
JINGJING WANG, Changsha University, Changsha, China
JIE WU, Department of Computer and Information Sciences, Temple University, Philadelphia, Pennsylvania, USA
KIAN-LEE TAN, School of Computing, National University of Singapore, Singapore, Singapore

In online social networks (OSNs), people usually join groups for communication. Information diffusion often occurs with some cost, either between individuals or within/among groups; and different opinions may compete with each other. The groups can make decisions based on the majority of the group members. This type of group consensus is common in group activities. However, existing research on maximization of competitive influence often neglects the effects of group consensus. To this end, we introduce the process of group consensus reaching in influence maximization and propose a novel Group consensus-based Competitive Linear Threshold (GCLT) propagation model; then we study the Budgeted Competitive Influence Maximization (BCIM) problem under the GCLT model. We reveal that the problem is NP-hard, and the objective function is proven to be neither submodular nor supermodular. To this end, we construct an equivalent Group consensus-based Competitive Live-Edge (GCLE) model of GCLT by sampling method. Based on GCLE, develop two submodular functions of the upper and lower bounds. Then, we propose the SBG algorithm by applying the Sandwich Approximation framework for the BCIM problem under the GCLT model. In SBG, we provide an approximate solution to the lower bound and the upper bound by the proposed OPIM-B algorithm. Then, we select the seed set of solutions that achieves the best influence spread in Monte Carlo simulations. We also propose two strategies to optimize SBG. The experiments on six real social network datasets verify the effectiveness and scalability of our method and validate the impact of group consensus on the competitive influence dissemination process, as well as the importance of considering the process of reaching group consensus.

CCS Concepts: • **Information systems** → **Social networks**; **Social advertising**;

Additional Key Words and Phrases: Group Consensus, Budgeted Influence Maximization, Competitive Influence Maximization, Linear Threshold

Associate Editor: Yu Yang

This research was supported by the National Natural Science Foundation of China (Grant No. 62172149), the Scientific Research Foundation of Hunan Provincial Education Department (24B0789), and the Changsha Natural Science Foundation (kq2502339).

Authors' Contact Information: Guobang Chen, College of Computer Science and Electronic Engineering, Hunan University, Changsha, China; e-mail: gbchen@hnu.edu.cn; Wenjun Jiang (corresponding author), College of Computer Science and Electronic Engineering, Hunan University, Changsha, China; e-mail: jiangwenjun@hnu.edu.cn; Kenli Li, College of Computer Science and Electronic Engineering, Hunan University, Changsha, China; e-mail: lkl@hnu.edu.cn; Jingjing Wang, Changsha University, Changsha, China; e-mail: wangjingjing2019@hnu.edu.cn; Jie Wu, Department of Computer and Information Sciences, Temple University, Philadelphia, Pennsylvania, USA; e-mail: jiewu@temple.edu; Kian-Lee Tan, School of Computing, National University of Singapore, Singapore, Singapore; e-mail: tankl@comp.nus.edu.sg.

ACM 1556-472X/2025/11-ART174
https://doi.org/10.1145/3768583

ACM Reference format:
Guobang Chen, Wenjun Jiang, Kenli Li, Jingjing Wang, Jie Wu, and Kian-Lee Tan. 2025. Integrating Group Consensus for Competitive Influence Maximization in OSNs. *ACM Trans. Knowl. Discov. Data.* 19, 9, Article 174 (November 2025), 38 pages.
https://doi.org/10.1145/3768583

1 Introduction

Nowadays, more and more people share their opinions, spread information, and propagate influence through interactions in **online social networks (OSNs)**. People usually participate in one or more groups with similar interests or hobbies in OSNs [55]. Furthermore, many offline groups, such as classes and grades in schools, also create online groups for the convenience of communication. These groups exist everywhere, and various kinds of information diffuse within and among groups. Moreover, human opinions and behaviors are easily influenced by their groups [16]. Therefore, comprehensively studying the effect of group consensus on the influence propagation process is essential to understanding people's behaviors and information diffusion patterns in OSNs.

Many studies have been done on information diffusion in OSNs, among which **influence maximization (IM)** is essential for many tasks such as campaigns or advertisements [22, 26]. The goal is to select some k nodes as the seed set to maximize the influence spread. Only a few works consider the group-level IM [52] or group IM [57, 58]. Yan et al. [52] proposed the problem of group-level IM with budget constraint and put forward a propagation model to calculate the influence spread (number of influenced individuals) of seed groups. Zhong and Guo [57] and Zhu et al. [58] proposed the Group IM problem to find the seed set such that the number of activated groups is maximized. However, group consensus hasn't been well studied in existing work. Moreover, all the above group-based works exploit the independent cascade propagation model; while there is relatively less research on the group-based IM under the **linear threshold (LT)** model, which is more natural for group IM [13].

According to the research on group decision-making [14, 16, 21], there is a relatively independent model of influence diffusion between members within the group, which we call the process of *group consensus reaching*. Zhang et al. [55] proposed new methods for measuring friendship closeness using the social identity theory, in which people tend to adopt the opinions or behaviors of users within the same group. Tran et al. [47] reviewed classical consensus approaches to group decision-making, classifying them based on different methods, including the reference domain, coincidence method, and individual centrality. They also introduced consensus models in group recommendation systems [11] and those in large-scale groups [37], which improve basic aggregation strategies and consider social relationship interactions. However, to our knowledge, the group consensus reaching process has not been fully studied in IM models, which may neglect the spread of influence within groups, i.e., it will not capture the impact of the group or its active members on its inactivated members, and vice versa.

Furthermore, opinion competition among several choices is prevalent within the groups. Therefore, it is necessary to consider competitors in the IM, which formulates the problem of **competitive influence maximization (CIM)**. Many works have been proposed to address this problem by extending the **influence cascading (IC)** model [20, 27]. They reveal the monotone and submodular CIM problem under the Competitive IC model. However, the CIM problem under the **competitive linear threshold (CLT)** model is proven to be neither submodular nor supermodular [5, 39]. Finally, there is usually an associated selection cost for each individual in the social network. It is very challenging to maximize the influence in such a competitive situation with a group structure and budget constraints.

Our Motivation. Keeping the above challenges in mind, our motivation is twofold: (1) integrating group consensus for CIM in OSNs and studying its effects on influence diffusion under the CLT model and (2) comprehensively studying the process and principles of influence diffusion in competitive situations with budget constraints, proposing effective diffusion models and IM algorithms considering group consensus reaching.

Our Contributions. We strive to study the impact of group consensus in the process of influence diffusion, better solve the problem of maximizing influence in group activities, and fully model the process of group consensus reaching. We propose a new propagation model based on the CLT model to study the maximization of competitive influence under budget constraints (**budgeted competitive influence maximization (BCIM)**). The main contributions are summarized in the following:

- We propose a novel **group consensus-based competitive linear threshold (GCLT)** model as the propagation model, which introduces the process of *group consensus reaching*. GCLT takes the group as a new l ink of influence spread between individuals and connects the process of influence propagation among the individuals and groups, making it more in line with the actual situation. Moreover, we study the properties of BCIM problem under the GCLT model and prove its NP-hardness, and the objective function is proven to be neither submodular nor supermodular (Section 3).
- To achieve a practical approximate solution, we construct an equivalent **group consensus-based competitive live-edge (GCLE)** model of GCLT by the sampling method. Based on GCLE, we redefine the objective function, confirm the optimal upper and lower bounds of the reverse reachable set, and develop the upper and lower bounds of the objective function (Section 4).
- Based on the refined upper and lower bound functions, we propose the SBG algorithm by applying the Sandwich Approximation framework for the BCIM problem under the GCLT model. We provide an approximate solution to the lower bound and the upper bound by OPIM-B algorithm. Then, we select the seed set of solutions that achieves the best influence spread in Monte Carlo simulations (Section 5).
- We conduct extensive experiments in real-world social networks. The results validate the effectiveness of our work, i.e., it can seek the most proper seed set to maximize activated nodes with a limited budget in a competitive environment. We also demonstrate that our algorithm can scale to million-scale networks. Moreover, we deeply study the influence diffusion process of the GCLT model, and analyze the influence of group consensus on the dissemination process of competitive influence. It shows the ability of the GCLT model to mine the potential connections (Section 6).

2 Related Work

We briefly review the related works in the literature and examine their connections with and differences from our work.

2.1 IM

Kempe and Kleinberg [22] studied the computational problem of maximizing influence and proved that the IM problem is NP-hard. They also proposed a greedy algorithm with an approximate ratio of $(1 - 1/e - \epsilon)$ to solve this problem. To improve the running time of the Greedy algorithm, Leskovec et al. [25] proposed CELF and CELF++ [18]. Moreover, Borgs et al. [4] proposed **reverse influence sampling (RIS)** algorithm for the IM problem. RIS returns a $(1 - 1/e - \epsilon)$ approximate solution with a probability of at least $1 - n^{-l}$. The main idea is to generate **reachable reverse (RR)** sets to estimate the objective function and use a greedy algorithm with a large enough RR set to

find the solution. Many methods including TIM/TIM++ [44], IMM [43], and SSA/DSSA [35] are all based on RIS.

There are some variants of the IM problem, for instance, **budgeted influence maximization (BIM)** [34], profit maximization [29], seed set minimization [59], location-aware IM [49], capacity constrained IM [54], fair IM [40], and so on. Moreover, some new techniques are exploited to address the IM problem. For example, Meena et al. [24] employed nature-inspired Cuckoo Search Optimization to solve the IM problem in dynamic networks. Theocharidis et al. [45] took advantage of the online Learning-to-Rank framework to study the problem of Adaptive Content-Aware IM, which tries to find k features to form a post in each round to maximize the cumulative influence of those posts over all rounds. Xie et al. [50] proposed to maximize influence via vertex countering, in which they studied the problem of influence countering, and proposed two novel algorithms with approximation guarantees and practical efficiency. Xue et al. [51] proposed a method for dynamically selecting key nodes to spread information rapidly under the graph burning model. Cai et al. [8] targeted a new research problem that aims to find l previous existing relationships that are estranged later, and reconnecting these relationships will maximize the expected influence spread by the given group in the future. Zhang et al. [56] proposed to improve the graph neural network with balanced IM.

The BIM problem was first proposed by Nguyen and Zheng [34]. In this problem, the seed set has to be chosen within a given budget to maximize the influence spread. They proposed a DAG heuristic with a $(1 - 1/\sqrt{e})$ factor approximation algorithm for this problem. Güney [15] developed the sample average approximation method for the BIM problem, which produced a near-optimal solution. Banerjee et al. [1] proposed a community-based solution framework and corresponding algorithm ComBIM for the BIM problem. Bian et al. [3] presented a framework *IMAGE*, in which an effective node selection strategy is proposed to alleviate the time complexity dependency on the size of the seed set, and an efficient approximate algorithm is devised for the BIM problem in large social graphs.

2.2 CIM

To better describe the propagation process in the real world, researchers add seed nodes with competitive relationships [9]. The two main directions are maximizing self-influence and minimizing the influence of competitors [7].

Bharathi et al. [2] first proposed the CIM problem by extending the IC model to a new propagation model. To maximize self-influence, Morozov [33] proposed a method to prevent the spread of erroneous information by ensuring that most users in social networks hear correct information before hearing bad one. To minimize the influence of competitors, Budak et al. [7] assumed that an activity C propagates bad information and is detected at the delay r, given the budget k, and the seed selection is organized to minimize the influence of C. He et al. [19] studied the influence blocking problem and established the CLT model based on the LT model. Its basic idea is to find a set of positive seeds with a maximum of k seeds, which minimizes the number of expected negative activation nodes. Borodin et al. [5] also extended the LT model in several ways to simulate the diffusion of competition effects. However, they only studied the problem of maximizing the influence, rather than the maximization of congestion [19].

2.3 Group-Based IM

A large social network is usually composed of several densely connected subsets, while only sparse links exist among different locally connected regions, which are called communities [36]. Influence calculation and community integration were studied early by Wang et al. [48]. They focused on mining the influential TOP-K nodes based on the community. They also introduced the idea of

Table 1. Notations

Notation	Description
$G(V, E, P, w, c)$	Directed graph G represents a social network with node set V, edge set E, and group set P
n, m	The number of nodes and edges
S_N, S_P	Competitor seed set, and positive seed set
$I(S_P)$	The influence spread of the seed set S_P
$U(S_P)$	The upper bound of the influence function $I(S_P)$
$L(S_P)$	The lower bound of the influence function $I(S_P)$
S_P^*, S_U^*, S_L^*	The optimal solution for maximizing $I(\cdot), U(\cdot), L(\cdot)$
OPT, OPT_U, OPT_L	$I(S_P^*), U(S_U^*), L(S_L^*)$
$n(C)$	Total number of nodes in group C
$\theta^+(v), \theta^-(v)$	The positive/negative activation threshold of node v
$\rho^+(C), \rho^-(C)$	The positive/negative group activation threshold of group C
$in(v), out(v)$	In degree/out degree neighbor nodes of node v
B	The budget constraint

community-based greedy algorithms and reduced the size of the entire network through the community. Song et al. [41] further improved [48] not only by adding parallel algorithms but also by considering the influence of nodes across the community to ensure accuracy. Some other works combined the community structure and influence to reduce the running time [1, 6, 30]. Moreover, Zhang et al. [55] introduced new measures for friendship closeness based on social identity theory, which describes that people tend to endorse the behaviors of users within the same group.

In other directions, Dai et al. [13] considered group effects and proposed a heuristic algorithm for the BCIM problem. Yan et al. [52] proposed the problem of group-level IM. They analyzed the influence relationship at the level of groups and selected seed groups to maximize influence. Zhu et al. [58] and Zhong and Guo [57] proposed the Group IM problem to find the seed set so that the number of activated groups is maximized. Meena et al. [31] addressed the diversification of activated nodes in the dynamic social network and tried to maximize the number of communities by utilizing bridge nodes.

2.4 Our Differences

While many researchers have addressed several key issues in IM, most existing works neglect the impact of the group consensus. Our work is different from others in two ways: (1) we integrate group consensus for competitive IM in OSNs and study its effects on influence diffusion, and (2) we comprehensively study the process and principles of influence diffusion in competitive situations with budget constraints, considering group consensus reaching.

3 Propagation Models and Problem Definition

In this section, we introduce the system settings and the Group consensus-based Competitive model considering group consensus. Table 1 displays the notations used in this article.

3.1 System Settings and Problem Formulation

Given a social network with groups, we represent it as a directed graph $G(V, E, P, w, c)$. Here, $V = \{v_1, v_2, ..., v_n\}$ is the set of nodes of graph G, and E is the set of edges among the nodes.

$P = \{C_1, C_2, ..., C_k\}$ is the set of groups of graph G, where $C \in P$ is the set of nodes of group C. And $\forall C_i, C_j \in P$ and $i \neq j$, $C_i \cap C_j = \emptyset$, $\bigcup_{i=1}^{k} C_i = V$. w is the edge weight function, i.e., each edge $e(u, v) \in E$ is associated with an influence weight $w(u, v) \in [0, 1]$. Each node $v \in V$ has a selecting cost $c(v) \geq 0$. For each node v, there is also a node to group weight $w(v, C)$ that $\sum_{v \in C} w(v, C) \in [0, 1]$ and $w(v, C) = 0$ for $v \notin C$.

In this section, we propose a propagation model based on group consensus effects, which are based on the LT model. Note that the idea of integrating group consensus can also be applied to other diffusion models, as long as they can consider group effects. Currently, we mainly focus on the LT model. In future work, it will be interesting to try other diffusion models. In the LT model, each node v has a threshold $\theta_v \in [0, 1]$. Given a seed set (i.e., the initial set of active nodes) $S \subseteq V$, the LT model works as follows. Let $S_t \subseteq V$ be the set of nodes activated at time t, with $S_0 = S$ and $S_t \cap S_{t-1} = \emptyset$. At time $t + 1$, each currently inactive node v becomes active if and only if $\sum_{active\ u \in in(v)} w(v, u) \geq \theta_v$, where $N(v)$ denotes the neighbor set of node v. The influence spread of S, denoted by $I(S)$, is the final number of activated nodes in the seed set S. The IM problem is to find such a seed set with k nodes while maximizing the influence spread $I(S)$. Moreover, in CIM, there are the competitor seed set S_N, and the positive seed set S_P, for which the influence spreads are denoted as $I(S_N)$ and $I(S_P)$ respectively, and their selection costs are denoted as $c(S_N)$ and $c(S_P)$, respectively. More notations are summarized in Table 1.

In this article, we study the BCIM problem, and it is formally described as follows.

Definition 1 (BCIM Problem). Given a directed graph $G(V, E, P, w, c)$, the budget B, and a set of competing nodes S_N, the BCIM problem is to find a positive seed set $S_P \subseteq V$ to maximize the influence spread $I(S_P)$ in a given influence propagation model with cost $c(S_P) = \sum_{v \in S_P} c(v) \leq B$, that is:

$$S_P^* = \underset{S_P \subseteq V,\ c(S_P) \leq B}{\arg\max} I(S_P). \tag{1}$$

Here, we use $I(S)$ to denote the influence spread in the propagation model.

To better describe the *group consensus reaching* process and follow the LT model, we use two linearly distributed influence factors $\rho, \theta \in [0, 1]$ to determine how easy it is for people and groups to accept an opinion in the real world. $\theta(v)$ is the influence threshold of nodes, and $\rho(C)$ is the active threshold of group C. Moreover, we use a parameter $\lambda \in [0, 1]$ to determine the acceptability of the group opinion for the members who hold other opinions or have no opinions after the group consensus is reached. For example, when people participate in a group buying activity, some may follow the group opinion, but the rest may not because the purchase behavior is not mandatory. In another case, when students in a class want to choose a place for group travel, they will be easily influenced by the opinions of the group. In this article, the parameter λ represents the probability that a node v accepts the group opinion.

In reality, the probability of accepting group opinion is determined by a combination of nodes, groups, social environments, and the opinions disseminated. In this article, we consider the parameter λ from a personal perspective. Incited by the measurements of friendship closeness in [55], we determine a personalized acceptance possibility λ for each individual within a group. That is, the acceptance probability $\lambda(v, C)$ of node v in a group C, is calculated following the user-group tightness in [55], as follows:

$$\lambda(v, C) = \frac{\sum_{u \in C \setminus v} w(u, v) \cdot sim(u, v)}{\sum_{u \in C \setminus v} w(u, v)}, \tag{2}$$

where $sim(u, v) \in [0, 1]$ is the similarity score of node v and u, which is defined as the cosine similarity of the embedded representations of Node2Vec [32]. Note that since we consider non-overlap groups in this article, each node only belongs to a group; thus, $\lambda(v, C)$ can also be simplified to be $\lambda(v)$.

It is worth noting that, in the above system settings, we mainly consider non-overlapping groups for simplicity. While in real-life scenarios, people may belong to different groups. The overlapping nodes that belong to different groups can be flexibly treated. For instance, if a node v belongs to k groups, we can calculate the acceptance probability $\lambda(v, C)$ to each group and update the state of v according to the state of the first group to reach v.

3.2 GCLT Model

We extend the LT model and consider the inter-group contact [38], group consensus [17], and competitors. Generally, the influence is spread all over the network, and the inter-group influence comes from contact between nodes of different groups. Therefore, we do not change the way that nodes spread influence between groups. However, the nodes may reach a consensus in a group and influence other nodes. Then, we take the group consensus effects into account and propose the GCLT model.

Definition 2 (GCLT Model). In a directed graph $G(V, E, P, w, c)$, for each group C, the GCLT model follows the CLT model and the group consensus reaching process to spread influence. Specifically, there are two disjoint seed sets, the positive seed set S_P and the competitor's (negative) seed set S_N. Each edge $e(u, v) \in E$ has two weights $w^+(u, v)$ and $w^-(u, v)$ that represent the influence weight of positive and negative diffusion, respectively. Every node has three possible states: *inactive*, *P_active*, and *N_active*, indicating that the node is not activated, activated by positive diffusion, or activated by negative diffusion, respectively. Each node v picks two independent thresholds $\theta^+(v)$, $\theta^-(v)$ uniformly from $[0, 1]$. The influence propagation of the GCLT model can be classified into two processes, i.e., group activation and individual node activation.

The two activation processes of influence propagation in the GCLT model are described as follows.

(1) Group activation of the GCLT model follows the CLT model and integrates the group consensus reaching process:
   — At step $t = 0$, $S_P^t = S_P$, $S_N^t = S_N$, where S_P^t and S_N^t are the *P_active* and *N_active* node set at step t.
   — At step $t \geq 1$, when the positive (or negative) influenced node weights of group C reach the group threshold $\rho(C)$, the group C will reach a consensus with the positive (or negative) diffusion. Then, each of the inactive and negative (or positive) node v in the group will become the same state as the group with probability $\lambda(v)$, and their states will remain unchanged. That is, group C becomes positively activated if

$$\sum_{u \in C \cap S_P^{t-1}} w^+(u, C) \geq \rho^+(C). \tag{3}$$

   Group C becomes negatively activated if

$$\sum_{u \in C \cap S_N^{t-1}} w^-(u, C) \geq \rho^-(C), \tag{4}$$

   where the $\rho^+(C)$ and $\rho^-(C)$ are the positive/negative activation threshold of group C, respectively. In the case when both Equations (3) and (4) are satisfied, we consider that C

becomes positively activated or negatively activated with equal probability. Note that the group activation is the only way to change the state of activated nodes.

(2) Individual node activation of the GCLT model follows the CLT model after the group activation in the same step:
   — At step $t \geq 1$, after the group activation process, positive influence and negative influence propagate independently as in the original LT model, using positive and negative weights/thresholds, respectively. Each *inactive* node v becomes *P_active* if

$$\sum_{u \in N(v) \cap S_P^{t-1}} w^+(u, v) \geq \theta^+(v) \tag{5}$$

   And node v becomes *N_active* if

$$\sum_{u \in N(v) \cap S_N^{t-1}} w^-(u, v) \geq \theta^-(v), \tag{6}$$

   where $N(v)$ denotes the neighbors of v in graph G. Similar to that of group activation, if v can be activated by both positive and negative diffusion, then v will randomly select an influence with equal probability.

The Differences between the GCLT Model and the LT Model. Unlike the LT model, there are two competing seed sets in the GCLT model: the positive seeds S_P and the negative seeds S_N. Consequently, each edge $e(u, v)$ is associated with two influence probabilities $w^+(u, v)$ and $w^-(u, v)$, and each node v has two independent thresholds θ_v^+ and θ_v^-. Then, a node can exist in one of three states: inactive, *P_active*, or *N_active*. Furthermore, the diffusion process of GCLT has two activation levels: the group activation and the node activation. When the cumulative positive (or negative) influence within a group exceeds its respective threshold, each node v within the group will transition to the same state with a probability of $\lambda(v)$, reflecting the consensus-building process of the group. Then, the remaining nodes undergo a node-level activation process similar to that of the classic LT model.

Figure 1 presents an example with 10 nodes and two groups C_1, C_2. Assume that $\lambda = 1$ and all propagation probabilities (i.e., edge weights) in this graph are $\frac{1}{d_{in}(v)}$ where $d_{in}(v)$ denotes the in-degree of node v. For nodes in group C_1, the positive thresholds θ^+ is 0 and negative thresholds θ^- is 1, and for nodes in group C_2, θ^+ is 1 and θ^- is 0. Assume that all group activation thresholds are $\frac{1}{2}$. The influence diffusion starts from the seed sets $S_N = \{v_4, v_5\}$ and $S_P = \{v_3\}$. At time $t = 1$, v_1, v_2 are activated by positive influence and v_6 is activated by negative influence. Then at $t = 2$, group C_1 is activated by positive influence and v_0 becomes *P_active*, and C_2 is activated by negative influence and nodes v_7, v_8, v_9 become *N_active*. Even though v_8 was *P_active*, it is influenced to follow the decision of the group.

3.3 Properties of BCIM Problem under GCLT Model

In this section, we first present the hardness of the BCIM problem under the GCLT model. Then we discuss the properties of the objective function $I(\cdot)$.

3.3.1 Hardness.

Theorem 3.1. *The BCIM problem under the GCLT model is NP-hard.*

Proof. Let's assume the cost of each node $c(v) = 1$, the competitor $S_N = \emptyset$, and each node is taken as a group. The BCIM problem under the GCLT model is then converted to the classical IM problem under the LT model. The IM problem has been proven to be NP-hard under the LT

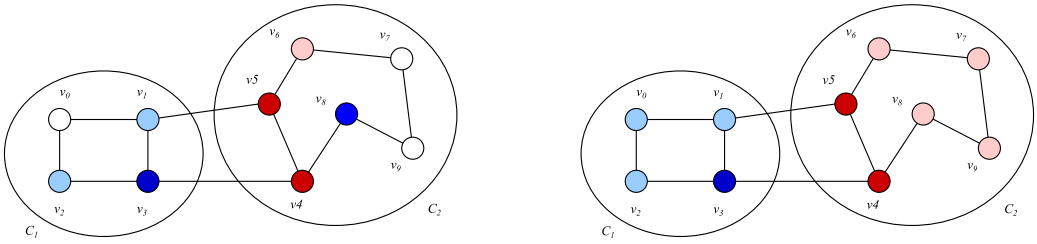


Fig. 1. Example network with 10 nodes and two groups. The deep red indicates $S_N = \{v_4, v_5\}$, dark blue indicates $S_P = \{v_3\}$, the light red and light blue indicate the nodes influenced by S_N and S_P, respectively.

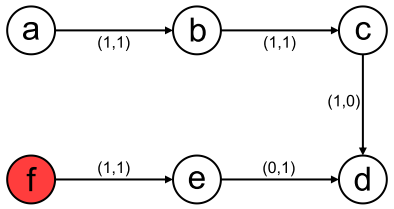


Fig. 2. A counter example.

model [22], and the BCIM problem is more complex for considering the budget constraint and the competition. Therefore, the BCIM problem under the GCLT model is also NP-hard. □

3.3.2 Modularity of Objective Function. Although the objective function in the IM problem under the LT model is monotone and submodular, the objective function $I(\cdot)$ in BCIM under the GCLT model is neither submodular nor supermodular.

For set $S \subseteq T \subset V$ and $x \in V \setminus T$, the function $I(\cdot)$ is submodular iff $I(S \cup \{x\}) - I(S) \geq I(T \cup \{x\}) - I(T)$. Correspondingly, $I(\cdot)$ is supermodular iff $I(S \cup \{x\}) - I(S) \leq I(T \cup \{x\}) - I(T)$.

Theorem 3.2. *$I(\cdot)$ is neither submodular nor supermodular in the BCIM problem under the GCLT model.*

Proof. This theorem will be proven by using the method of contradiction. First, we formulate a counter example (Figure 2). Consider an instance of IM problem $G = (V, E)$ with $V = \{a, b, c, d, e, f\}$, $E = \{(a, b), (b, c), (c, d), (f, e), (e, d)\}$. The positive and negative weight of each edge is shown as a tuple $(w^+(e), w^-(e))$. For simplicity, we treat each node as a group. Let $S_N = \{f\}$, we have $I(\emptyset) = 0$, $I(\{a\}) = 3$, $I(\{a, c\}) = 4$, $I(\{c\}) = 2$, $I(\{a, e\}) = 5$, and $I(\{e\}) = 1$. Therefore, $I(\{a, e\}) - I(\{a\}) = 2 > I(\{e\}) - I(\emptyset) = 1$, which means $I(\cdot)$ is not submodular. On the other hand, $I(\{a, c\}) - I(\{c\}) = 2 < I(\{a\}) - I(\emptyset) = 3$, which means $I(\cdot)$ is not supermodular. □

Due to non-submodular and non-supermodular properties in the BCIM problem under the GCLT model, we cannot use the naive greedy algorithm on the objective function to get an optimum solution within a factor of $(1 - 1/\sqrt{e})$. So we construct an equivalent GCLE model of GCLT by sampling method (Section 4) and design a framework that solves the optimization problems on the lower and upper bound functions to obtain a solution that can be bounded (Section 5).

4 GCLE Model

In this section, we construct an equivalent GCLE model of GCLT by sampling method. Based on GCLE, we redefine the objective function, confirm the optimal upper and lower bounds of the reverse reachable set, and develop the upper and lower bounds of the objective function.

As mentioned in Section 3, the objective function is neither submodular nor supermodular. Hence, we cannot use the naive greedy algorithm to get an approximation guarantee. Kempe and Kleinberg [22] showed that the LT model can be equivalently represented by the "live-edge" model. Following the method in [19] and [39], we will construct a modified GCLE model, which is equivalent to the GCLT model.

Moreover, the Sandwich method [28] provides a solution framework to guarantee the theoretical approximation for the non-submodular objective function optimization problem. There are three key steps: (1) find the submodular upper and lower bounds of the objective function, (2) propose two different solutions by solving the optimization problems on the upper and lower bound functions, and (3) compare the effects of the two solutions on the original objective function and take the better one as the solution to the original problem. We will exploit the Sandwich framework for the BCIM problem in the GCLT model. Hence, based on the GCLE model, we will estimate the upper and lower bounds of the objective function. The details are as in the following subsections.

4.1 Construction of GCLE Model

In the graph $G = (V, E)$ under the GCLT model, we cannot sample a random graph with a computable sample probability, because the thresholds of groups are continuously distributed. To solve this problem, we add a group node x_i to the graph to represent the group C_i and connect each node $v \in C$ to the group node with weight $w(v, x_i) = w(v, C_i)$. Then, we connect the node x_i to each node v in this group with weight $w(x_i, v) = \lambda(v)$.

Assume X is the group node set and E' is the edge set between the nodes in V and the nodes in X as selected above. We construct a sample graph g from $G = (V \cup X, E \cup E')$ as follows.

We sample a graph g_P with positive diffusion. First, for each $v \in V$, we randomly select one in-edge $e(u, v) \in E$ with probability $w^+(u, v)$ and do not select any in-edge in E with probability $1 - \sum_{u \in N(v)} w^+(u, v)$. For any group node $x \in X$, GCLE randomly selects an incoming edge with the probability of $w^+(u, x)$; and for each outgoing edge $e(x, v) \in E'$, GCLE selects this edge with the probability of $\lambda(v)$, i.e., the acceptance probability of node v to its group. The selected edge is called the P-live edge and is denoted as E_P. Except for the P-live edges obtained by the above method, all other edges are deleted to obtain a sample graph g_P with positive diffusion. Meanwhile, we also randomly sample a propagation g_N in the same way. The selected edge is called the N-live edge and is denoted as E_N. Finally, we construct the sample graph g as the union of g_P and g_N.

In graph g, we denote $M_P^t \subset X$ and $M_N^t \subset X$ as sets of *P_active* group nodes and *N_active* group nodes on g at step t, respectively. The distribution of *P_active* and *N_active* nodes in g happens in discrete step t and is as follows:

- At step $t = 0$, $S_P^t = S_P$, $S_N^t = S_N$, $M_P^t = \emptyset$, and $M_N^t = \emptyset$.
- At step $t \geq 1$, a group node $x \in X \setminus (M_P^{t-1} \cup M_N^{t-1})$ becomes *P_active* if x is reachable from S_P^{t-1} in one step in g_P, but not reachable from S_N^{t-1} in one step in g_N. If this is the case, then add x to M_P^t. Symmetrically, if x is reachable from S_N^{t-1} in one step in g_N but not reachable from S_P^{t-1} in one step in g_P, then add x to M_N^t.
- At step t, after activation of group nodes, a node $v \in V$ becomes *P_active* if v is reachable from M_P^t in one step in g_P or becomes *N_active* if v is reachable from M_N^t in one step in g_N. The activation of group nodes and the nodes connected from group nodes is simultaneous.

— At step t after group nodes activate the nodes, an *inactive* node $v \in V$ becomes *P_active* if v is reachable from S_P^{t-1} in one step in g_P, but not reachable from S_N^{t-1} in one step in g_N. Symmetrically, v becomes *N_active* if v is reachable from S_N^{t-1} in one step in g_N, but not reachable from S_P^{t-1} in one step in g_P.
— If at step $t \geq 1$, a node $v \in V \cup X$ is reachable from S_P^{t-1} in one step in g_P and reachable from S_N^{t-1} in one step in g_N, v becomes *P_active* or *N_active* with equal probability 1/2.

Theorem 4.1. *For a given positive set S_P and negative set S_N, the distribution of* P_active *nodes and* N_active *nodes at each step t on GCLT model and GCLE are equivalent.*

The proof of Theorem 4.1 is presented in Appendix A.

Assume $\mathcal{G}$ contains all possible realizations of G. Let $\Pr[g]$ be the probability that g can be generated from G, then we have:

$$\Pr[g] = \lambda^l (1-\lambda)^{n-l} \Pr[g_P] \cdot \Pr[g_N],$$

where

$$\Pr[g_P] = \prod_{e \in E_P \setminus E_l} w^+(e) \prod_{v \in \overline{V}} \left(1 - \sum_{u \in N(v)} w^+(u,v)\right),$$

$$\Pr[g_N] = \prod_{e \in E_N \setminus E_l} w^-(e) \prod_{v \in \overline{V}} \left(1 - \sum_{u \in N(v)} w^-(u,v)\right).$$

$\Pr[g_P]$ and $\Pr[g_N]$ are the probabilities that g_P and g_N can be generated. $\overline{V}$ is the set of nodes without selecting any in-edge in E. E_l is the set of edges from group nodes to origin nodes. And $l = |E_l|$ is the number of selected edges from group nodes to origin nodes.

We denote $\gamma_g(v|S_P)$ as the influence on node v in sample graph g, which is defined as follows:

$$\gamma_g(v|S_P) = \begin{cases} 1, & \text{if } v \text{ is } P\_active \text{ after propagation in } g \\ 0, & \text{otherwise} \end{cases}. \tag{7}$$

Node v is called the source node. Let S_P be the set of randomly selected nodes in $V \setminus S_N$ and g be a random graph generated from G, we have $I(S_P) = n \cdot \mathbb{E}[\gamma(S_P)]$, where $\gamma(S_P)$ is the expectation of $\gamma_g(v|S_P)$ over all random source nodes and sample graphs.

In the next subsections, we will leverage the equivalence between the GCLT and the GCLE models to design the lower and upper bounds of the objective function.

4.2 Formulation of the Upper Bound

In this section, we will formulate the optimal submodular upper bound for the objective function. The main idea is that, in a possible world g, we choose the node u that can make v becomes *P_active* with probability $P_g(u,v) > 0$. We denote the selected node set as $OPT_{R_U} = \{u | P_g(u,v) > 0\}$ because it contains all the possibly reachable nodes to v without any unreachable nodes.

Then we use the following lemma to give the sufficient and necessary conditions for $P_g(u,v) > 0$.

Lemma 4.1. *The sufficient and necessary conditions for $P_g(u,v) > 0$ are:*

(1) There exist $S_P \subseteq V \setminus S_N$ with $\gamma_g(v|S_P) = 0$ and $\gamma_g(v|S_P \cup \{u\}) = 1$ and
(2) Node v is positively activated through a path from u to v.

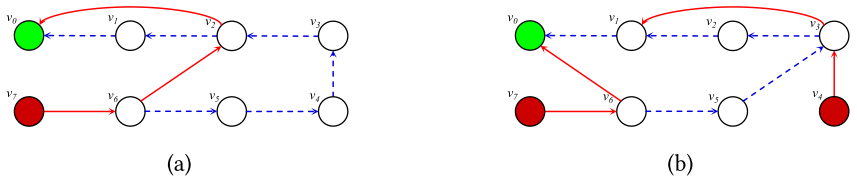


Fig. 3. Examples of sample graphs with target node v_0, and (a) $S_N = \{v_7\}$, (b) $S_N = \{v_4, v_7\}$, the red solid lines and blue dotted lines indicate the edges in g_N and g_P, respectively.

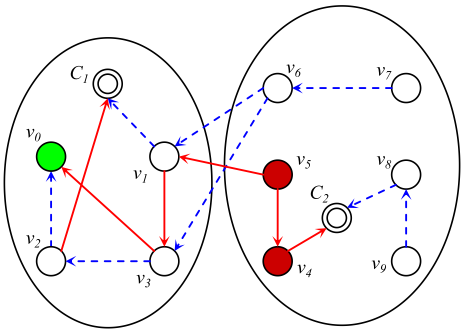


Fig. 4. A sample graph g with target node v_0, $S_N = \{v_4, v_5\}$, two groups $\{C_1, C_2\}$, the edges from group to the nodes are omitted, the red solid lines and blue dotted lines indicate the edges in g_N and g_P, respectively.

Condition 1 specifies that the node u is influential, and condition 2 guarantees that the positive influence comes from node u rather than other positive seeds. Figure 3 gives two examples. In Figure 3(a), $OPT_{R_U} = \{v_0, v_1, v_2, v_3, v_4, v_6\}$ and the source node is v_0. Node v_5 is not reverse reachable because v_7 reaches v_2 faster than v_5, and if v_6 (or v_2) is selected to block the influence of v_7, node v_0 will be positively activated. In Figure 3(b), $OPT_{R_U} = \{v_0, v_1, v_2, v_3, v_4, v_5\}$. The nodes in OPT_{R_U} can influence v_0 when $S_P = v_6$. However, even if node v_6 meets condition 1 (e.g., $S_P = \{v_3\}$), node v_0 is activated by the nodes in the path before node v_6, and condition 2 is not met.

In the GCLE model, each node is connected by at most a group node and at most one other node. The node in the GCLE model can be influenced in two directions. Therefore, it is difficult and time-consuming to form a set OPT_{R_U}. To formulate an upper bound, we need to select a reverse reachable node set R such that $OPT_{R_U} \subseteq R$ and undergo the following method.

For a random source node v and a sample graph g with a given negative seed set S_N, we can find the node set R that for each node $u \in R$, u may influence node v satisfying the following conditions: There is a path from u to v in g_P without nodes from S_N, and there is not a node t in the path can be influenced by S_N in one step but cannot be influenced by u in one step.

Figure 4 shows an example of a sample graph g. In this example, the sample graph g contains 10 nodes and two group nodes (i.e., groups) $\{C_1, C_2\}$. The source node is v_0 and $S_N = \{v_4, v_5\}$. Otherwise, g_P contains the blue dashed edges, and g_N contains the red solid edges. Let $\lambda = 1$, we have $R = \{v_0, v_1, v_2, v_3, v_6, v_8\}$. Nodes v_0, v_2, v_3, v_4 lie on the simple path that ends at v_0 and are not blocked by S_N. Node v_1 can influence v_0 by influencing the group node C_1. v_6 can influence v_3 to positively activate node v_0. v_8 can indirectly block v_5 by activate group node C_2 and then activate node v_6. v_7 may be activated by node v_4 through C_2. We call the set R generated above the *Upper bound Reversal Reachable* (*URR*) set.

For any $S_P \subseteq V \setminus S_N$ and URR set $R_g(v)$ with sample graph g and source node v, denote the coverage:

$$f_g(v|S_P) = \begin{cases} 1, \text{ if } R_g(v) \cap S_P \neq \emptyset \\ 0, \text{ otherwise} \end{cases}. \tag{8}$$

Lemma 4.2. *For any set $S_P \subseteq V \setminus S_N$, a random source node v, and random sample graph g, there is $f_g(v|S_P) \geq \gamma_g(v|S_P)$.*

If we define the upper bound function as $U(S_P) = n \cdot \mathbb{E}[f(S_P)]$ and $\mathcal{R}$ is a set of URR sets, we can estimate $U(S_P)$ as

$$\hat{U}(S_P) = \frac{n}{|\mathcal{R}|} \sum_{R \in \mathcal{R}} f(S_P). \tag{9}$$

The upper bound function $U(\cdot)$ is monotone and submodular, since $U(\cdot)$ is a form of weight coverage function.

Theorem 4.2. *Given an instance of the GCLT model $G = (V, E, P)$, $U(\cdot)$ is an optimal upper bound of $I(\cdot)$, that is $U(S_P) \geq I(S_P)$ for any node set $S_P \subseteq V \setminus S_N$.*

Proof. Using Lemma 4.2, we have

$$\begin{aligned} I(S_P) &= n \cdot \sum_{g \in \mathcal{G}} \Pr[g] \sum_{v \in V} \gamma_g(v|S_P) \\ &\leq n \cdot \sum_{g \in \mathcal{G}} \Pr[g] \sum_{v \in V} f_g(v|S_P) = U(S_P). \end{aligned} \tag{10}$$

□

We define $GN(v)$, the group node of v, as follows:

$$GN(v) = \begin{cases} x, \ v \in V \text{ and } x \text{ is the selected group node with probability } \lambda \\ v, \ v \in X \end{cases}. \tag{11}$$

Then, we design an algorithm to generate a URR set. This is shown in Algorithm 1. We first generate the group node set X according to the group distribution P and randomly select a node $v \in V$. Next, we use the BFS algorithm to find the nodes that can influence node v. We say a node u is an *indirect blocking node* if u or $GN(u)$ can be activated by S_N in one step. If $GN(u)$ can be activated by S_N in one step, then the neighbor nodes of u cannot influence u. When no new nodes can be selected, the algorithm stops.

4.3 Formulation of the Lower Bound

In this subsection, we formulate a lower bound submodular function for the objective function. The main idea is that, in a possible world g, we choose all the nodes u that can make v become *P_active* with probability $P_g(u, v) = 1$. We denote the selected node set as $OPT_{R_L} = \{u | P_g(u, v) = 1\}$.

The key to judging whether $P_g(u, v) = 1$ is stated in the following lemma.

Lemma 4.3. *The sufficient and necessary conditions for $P_g(u, v) = 1$ is: For a random target node $v \in V$ in a sample graph g, $\forall S_P \subseteq V \setminus S_N$, $\gamma_g(v|S_P \cup \{u\}) = 1$, where $u \in V \setminus (S_P \cup S_N)$.*

Lemma 4.3 shows that if v can be activated by seed set $\{u\}$ in a sample graph g, then $P_g(u, v) = 1$, according to the monotonicity of the objective function $I(\cdot)$. It also means for each node t in the path from a node u to v in g_P, u reaches the node t earlier than S_N or blocks the influence path from

Algorithm 1: URR Set Generation Algorithm (URR)

Input: Graph $G(V, E, P, w, c)$, negative seed set S_N
Output: A URR set R

Generate group node set X according to the group set P
Randomly select a node $v \in V$
Edge set $g_P \leftarrow \emptyset$, $g_N \leftarrow \emptyset$, Node set $R \leftarrow \emptyset$, FIFO queue $Q \leftarrow \{v\}$
while $Q \neq \emptyset$ **do**
    $v \leftarrow first\ node\ in\ Q$, $Q \leftarrow Q - \{v\}$, $block \leftarrow FALSE$
    if $\nexists$ in-edge (u, x) in g_P for $x = GN(v)$ **then**
        Select a positive in-edge (u_P, x) and a negative in-edge (u_N, x)
        Add edge (u_P, x) to g_P and add edge (u_N, x) to g_N
        if $u_P \notin S_N \cup R$ **then**
            $Q \leftarrow Q + \{u_P\}$
    if $(u_N, x) \in g_N$ and $u_N \in S_N$ **then**
        $block \leftarrow TRUE$
    if $v \notin S_N \cup R$ **then**
        $R \leftarrow R + \{v\}$
        Select a positive in-edge (u_P, v) and a negative in-edge (u_N, v)
        if $u_N \notin S_N$ and $u_P \notin R$ and $block = FALSE$ **then**
            $Q \leftarrow Q + \{u_P\}$
        if $u_N \in S_N$ and $u_P \notin R$ and $block = FALSE$ **then**
            $R \leftarrow R + \{u_P\}$
return R

S_N to t. Taking Figure 3(a) as an example, node v_6 reaches v_0 later than node v_7, but v_6 can block the influence from v_7 and influence v_0. Specially, for the source node v, if $GN(v)$ can be influenced by S_N, the nodes cannot influence v through a path only containing nodes $t \in V$ unless the negative influence can be blocked by u, because S_N can cover and eliminate positive influence by negatively influencing $GN(v)$.

According to Lemma 4.3, we need a global simulation in a sample graph g to confirm if $P_g(u, v) = 1$, which takes a lot of computation time. Instead of that, we can find a node set $R \subseteq OPT_{R_L}$ to formulate a suboptimal lower bound, satisfying:

(1) If there is not a path from S_N to $GN(v)$ in g_N, there is a path from u to v in g_P, and u can influence the nodes in the path faster than all nodes in S_N.
(2) Or, if there is a path from S_N to $GN(v)$ in g_N, there is a path from u to v in g_P, and u can influence the nodes in the path faster than all nodes in S_N and block the influence paths from S_N to $GN(v)$ in g_N.

We do not consider all the nodes that can both block S_N and influence target node v because it is complex. To find the nodes that can block the negative influence to $GN(v)$ and can spread the positive influence to v, we can first formulate a set A with nodes that can block the negative influence to $GN(v)$, then formulate the RR set B of node v that can influence v considering $GN(v)$ is never negatively influenced. The set $A \cap B$ is the result.

Taking Figure 5 as an example, we have $R = \{v_1, v_3, v_6, v_8\}$. Nodes v_3, v_6 lie on the path that ends at C_1 and are not blocked by or later than S_N. Node v_8 can influence C_2 and then influence node v_6. Nodes v_0, v_1, v_2 are in the path to v_0. Group node C_1 is reachable from S_N, but node v_1 can block the negative influence. So, v_1 is reachable but v_0 and v_2 are not. We call the set R generated above *Lower bound Reachable Reversal (LRR) set*.

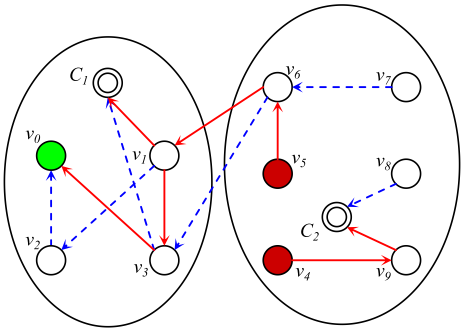


Fig. 5. A sample graph g with target node v_0, $S_N = \{v_4, v_5\}$, two groups $\{C_1, C_2\}$, the red solid lines and blue dotted lines indicate the edges in g_N and g_P, respectively.

For any $S_P \subseteq V \setminus S_N$ and an LRR set $R_g(v)$ with sample graph g and source node v, denote the coverage:

$$h_g(v|S_P) = \begin{cases} 1, & \text{if } R_g(v) \cap S_P \neq \emptyset \\ 0, & \text{otherwise} \end{cases}. \tag{12}$$

Lemma 4.4. *For any set $S_P \subseteq V \setminus S_N$, a random source node v and random sample graph g, there is $h_g(v|S_P) \leq \gamma_g(v|S_P)$.*

If we define the lower bound function $L(S_P) = n \cdot \mathbb{E}[h(S_P)]$, let $\mathcal{R}$ be a set of LRR sets, we can estimate $L(S_P)$ as

$$\hat{L}(S_P) = \frac{n}{|\mathcal{R}|} \sum_{R \in \mathcal{R}} h(S_P). \tag{13}$$

It is obvious that the lower bound function $L(\cdot)$ is monotone and submodular.

Theorem 4.3. *Given an instance of GCLT model $G = (V, E, P)$, $L(\cdot)$ is a lower bound of $I(\cdot)$, that is $L(S_P) \leq I(S_P)$ for any node set $S_P \subseteq V \setminus S_N$.*

Algorithm 2 shows the generation of the LRR set. We first generate a group node set X and randomly select a node v as the source node. Then, we will see if the group of v (i.e., $GN(v)$) is reachable from S_N. If it is not, we generate the LRR set using the LRR set generation procedure (LRR-P), shown in Algorithm 3. If the group of v will be activated by S_N, we first generate the node set R_b that can block the influence from S_N to $GN(v)$. We use a kind of DFS algorithm to generate R_b by judging if a node can reach the nodes in the negative path faster than S_N nodes, using the LRR-P Algorithm. We use V_b to record the blocked nodes in the previous traversals. If a group node in the negative path is not blocked and can be activated by S_N, we will also ensure the group is blocked. Therefore, we use R_c as the candidate block node set in the traversal procedure.

In Algorithm 3, we use the DFS approach to find the reachable nodes with probability 1. First in lines 2–5, if the source node is an original node, we choose the group node $GN(v)$ and use the distance from S_N to the nodes to update t_{max} which represents the permissible maximum time for a positive node to reach the source node. Then, we generate and update the reachable node set to $GN(v)$. In lines 6 and 7, we add the node v to R if v is not a negative node and the reverse diffusion steps $t_{max} - t$ from the current node v is bigger than the previous reverse diffusion steps

Algorithm 2: LRR Set Generation Algorithm (LRR)

Input: Graph $G(V, E, P, w, c)$, negative seed set S_N
Output: An LRR set R

Generate group node set X and edge set E' according to the group set P
Randomly select a node $v \in V$
Edge set $g_P \leftarrow \emptyset$, $g_N \leftarrow \emptyset$
$R \leftarrow \emptyset$, $t \leftarrow 0$, $t_{max} \leftarrow n$
Blocked node set $V_b \leftarrow \emptyset$, $x \leftarrow GN(v)$
if x is reachable from S_N **then**
    Influence block node set $R_b \leftarrow \emptyset$
    Candidate block node set $R_c \leftarrow V$
    while $x \notin V_b$ and $x \notin S_N$ and $R_c \neq \emptyset$ **do**
        $V_b \leftarrow V_b \cup \{x\}$
        if $GN(x) \notin V_b$ and $GN(x)$ is reachable from S_N **then**
            $R_c \leftarrow$ influence block node set for node $GN(x)$
            $V_b \leftarrow V_b \cup \{GN(x)\}$
        $R_b \leftarrow R_b \cup (\text{LRR-P}(G(V + X, E + E', w, c), R_b, S_N, g_P, g_N, t, t_{max}, x) \cap R_c)$
        Select from g_N if exists or randomly select in-edge (u, x)
        $x \leftarrow u$
    $R \leftarrow \text{LRR-P}(G(V + X, E + E', w, c), R, S_N, g_P, g_N, t, t_{max}, v) \cap V$
    $R \leftarrow R \cap R_b(v)$
else
    $R \leftarrow \text{LRR-P}(G(V + X, E + E', w, c), R, S_N, g_P, g_N, t, t_{max}, v) \cap V$
return R

Algorithm 3: LRR Set Generation Procedure (LRR-P)

Input: Graph $G(V + X, E, w)$, R, S_N, g_P, g_N, max influence steps t_{max}, source node v
Output: LRR set R

while $v \notin R$ and $t < t_{max}$ **do**
    if $v \notin X$ **then**
        $x \leftarrow GN(v)$
        $t_{max} \leftarrow Min(t_{max}, t + d(SN, x))$
        $R \leftarrow \text{LRR-P}(G, R, S_N, g_P, g_N, t, t_{max}, x)$
    if $v \notin S_N$ and $t_{max} - t > R_t(v)$ **then**
        $R \leftarrow R + \{v\}$ and update $R_t(v)$
        $t_{max} \leftarrow Min(t_{max}, t + d(SN, v))$
        Select from g_P if exists or randomly select in-edge (u, v)
        $v \leftarrow u$
    $t \leftarrow t + 1$
return R

$R_t(v)$ when v is selected ($R_t(v) = +\infty$ if never selected). Next, we update the permissible maximum time and randomly select a reachable node u for the next loop.

4.4 Discussion on the General Application

This section defines the optimal upper and lower bound reverse reachable sets under the GCLT model based on the GCLE model, which can be easily applied to the general CLT model. The reverse reachable set sampling method proposed in this section can also be extended to the Group IM problem. We only need to ensure that the source nodes are group nodes for reverse sampling.

Algorithm 4: Budgeted Maximum Coverage procedure (MaxCov-Budget)

Input: RR set $\mathcal{R}$, negative seed set S_N, budget B
Output: Seed set S_P

$S_P \leftarrow \emptyset, U \leftarrow V \setminus S_N$;
while $U \neq \emptyset$ **do**
  $v \leftarrow \arg\max_{u \in U} \mathrm{Cov}_{\mathcal{R}}(u|S)/c(u)$
  if $c(v \cup S) \leq B$ **then** $S \leftarrow S + \{v\}$
  $U \leftarrow U \setminus \{v\}$
$v_{max} \leftarrow \arg\max_{u \in U, c(u) < B} \mathrm{Cov}_{\mathcal{R}}(u)$
$S_P \leftarrow \arg\max_{S' \in \{S, \{v_{max}\}\}} \mathrm{Cov}_{\mathcal{R}}(S')$
return S_P

5 SBG: Group Consensus-Based Algorithm for BCIM Problem in GCLT Model

This section introduces our group consensus-based algorithm *SBG* for the BCIM problem in the GCLT model. We first propose an algorithm to find maximum coverage with a budget constraint (Section 5.1). Then, we exploit the sandwich approximation strategy proposed by Lu et al. [28], which analyzes the objective function by comparing its upper and lower bounds. To be specific, for the problem of maximizing the upper bound $U(\cdot)$ and the lower bound $L(\cdot)$, we extend the OPIM-C [42] algorithm with a Budget constraint and propose the OPIM-B algorithm in Section 5.2, which obtains solutions based on the URR and LRR set generation methods. Based on the two sets, we exploit the sandwich approximation framework and choose the seed set of solutions that achieves the best influence spread in Monte Carlo simulations with a theoretical guarantee (SBG in Section 5.3). Next, we propose two optimization strategies (Section 5.4) and analyze the complexity of the algorithms (Section 5.5).

5.1 MaxCov-Budget: Finding Maximum Coverage with Budget Constraint

In this section, we design Algorithm 4 to illustrate the greedy Budgeted Maximum Coverage procedure (MaxCov-Budget) to find a coverage set with an approximation ratio of $(1 - 1/\sqrt{e})$ [23], where $\mathrm{Cov}_{\mathcal{R}}(S_P) = \sum_{R \in \mathcal{R}} \min\{|S_P \cap R|, 1\}$.

The iterative procedure adds a node to S_P. In each iteration, a node v is first identified that maximizes the coverage gained $\mathrm{Cov}_{\mathcal{R}}(u|S) = \mathrm{Cov}_{\mathcal{R}}(S + \{u\}) - \mathrm{Cov}_{\mathcal{R}}(S)$ over the cost ratio. Then, if the cost of node v is less than the remaining budget, it will be placed in the seed set S_P. Next, select the node v_{max}, which has the highest coverage, compare the coverage of S and v_{max}, and output the one with the highest coverage.

Algorithm 4 can be implemented in linear time in terms of the total size of the RR sets. Therefore, the complexity is related to the generated RR sets.

5.2 OPIM-B: Polling-Based Greedy Algorithm with Budget Constraint

In this section, we extend the OPIM-C [42] algorithm to solve the BCIM problem based on the Budgeted Maximum Coverage procedure in Algorithm 4. We will introduce the main idea of OPIM-B (Algorithm 5) and clarify its difference from OPIM-C, followed by its implementation and analysis. Note that we conduct the discussion mainly based on the upper bound $U(\cdot)$, and the process for the lower bound $L(\cdot)$ is similar.

5.2.1 The Main Process of OPIM-B. For a random RR set $R \in \mathcal{R}$, spread $U(\cdot) = n \cdot \Pr[R \cap S_P \neq \emptyset]$. The objective in the upper bound can be solved by finding the seeds S_P with the maximum coverage $\mathrm{Cov}_{\mathcal{R}}(S_P)$ in $\mathcal{R}$. Just as shown in Section 5.1.

Algorithm 5: OPIM-Based Method for BIM (OPIM-B)

Input: Graph $G(V, E, P, w, c)$, budget B, and $\epsilon, \delta \in (0, 1)$
Output: A $(1 - 1/\sqrt{e} - \epsilon)$-optimal solution, S_P.

1: Initialize Λ_{max} by Eq.(16); $\Lambda_0 = \frac{\epsilon^2 I_1^g(S)}{n} \cdot \Lambda_{max}$; $i_{max} \leftarrow \lceil \log_2 \frac{\Lambda_{max}}{\Lambda_0} \rceil$
2: Generate $\mathcal{R}_1$ and $\mathcal{R}_2$ with $|\mathcal{R}_1| = |\mathcal{R}_2| = \Lambda_0$ by RIS
3: **for** $i \leftarrow 1$ **to** i_{max} **do**
4: $S_P \leftarrow$ MaxCov-Budget$(\mathcal{R}_1, B)$
5: Compute $\sigma^l(S_P)$and $\sigma^u(S^o)$ by Eq. (19) and Eq. (20), respectively, with $\delta_1 = \delta_2 = \delta/(3i_{max})$
6: **if** $\frac{\sigma^l(S_P)}{\sigma^u(S^o)} \geq 1 - 1/\sqrt{e} - \epsilon$ **or** $i = i_{max}$ **then return** S_P;
7: Double the sizes of $\mathcal{R}_1$ and $\mathcal{R}_2$ with new random RR sets;

Let $\sigma(\cdot) = U(\cdot)$ (resp. $\sigma(\cdot) = L(\cdot)$), the pseudocode of OPIM-B is illustrated in Algorithm 5. Initially, OPIM-B defines the constants Λ_{max} and Λ_0 (Line 1), representing the worst case and the initial number of random RR sets, respectively. Then it constructs two sets $\mathcal{R}_1$ and $\mathcal{R}_2$, both containing Λ_0 random RR sets (Line 2). After that, it runs in an iterative manner to verify if the selected seed pairs have satisfied the approximation guarantee with at most Λ_{max} random RR sets. In each iteration, it first invokes MaxCov-Budget by using $\mathrm{Cov}_{\mathcal{R}}(\cdot)$ as the evaluation function and selects the set S_P (Line 4). To verify whether the selected S_P provides the desired approximation guarantee, it calculates the lower bound $\sigma^l(S_P)$ of S_P's expected spread and the upper bound $\sigma^u(S^o)$ of optimal seed set S^o's expected spread using $\mathcal{R}_1$ and $\mathcal{R}_2$ (Line 5). OPIM-B terminates and returns the S_P if it reaches the i_{max}th iteration or

$$\frac{\sigma^l(S_P)}{\sigma^u(S^o)} \geq 1 - 1/\sqrt{e} - \epsilon. \tag{14}$$

Otherwise, it doubles the sizes of $\mathcal{R}_1$ and $\mathcal{R}_2$ and continues (Lines 6 and 7).

In order to ensure the correctness of OPIM-B, we modify the following aspects compared to OPIM-C: (i) a new Λ_{max} is provided to ensure the approximation guarantee in the worst case; (ii) $\sigma^u(S^o)$ and $\sigma^l(S_P)$ are recomputed to ensure the approximation.

In the sampling process, the most important task is to determine the number of samples needed to satisfy the given estimation error. Based on the RIS process, Tang et al. [43] proved the $(1 - 1/e)$-approximation guarantee of the RIS-based algorithm for the IM problem with the following sample size:

$$|\mathcal{R}| \geq \frac{2n\left((1 - 1/e)\sqrt{\ln \frac{2}{\delta}} + \sqrt{(1 - 1/e)(\ln \binom{n}{k} + \ln \frac{2}{\delta})}\right)^2}{\epsilon^2 \cdot \mathrm{OPT}}. \tag{15}$$

However, this sample size is not applicable to the BIM problem. Then, we show the number of random RR sets, which ensures Algorithm 4 returns a $(1 - 1/\sqrt{e} - \epsilon)$-approximation solution with probability of at least $1 - \delta$, is

$$\Lambda_{max} = \frac{2n\left((1 - 1/\sqrt{e})\sqrt{\ln \frac{2}{\delta}} + \sqrt{(1 - 1/\sqrt{e})(\ln \left(\binom{n}{k_{max}} + n\right) + \ln \frac{2}{\delta})}\right)^2}{\epsilon^2 I_1^g(S)}. \tag{16}$$

In Equation (16), k_{max} is the maximum size of a seed set and $I_1^g(S)$ is the expected influence spread of groups activated by a random seed set S in the first spread process, as follows:

$$I_1^g(S) = \sum_{v \in S, C \in P} \left[w^+(v, C) \cdot \left(1 - \sum_{u \in S_N} w^-(u, C) \right) \cdot \sum_{u \in C} \lambda(u) \right]. \tag{17}$$

Another difference between our *OPIM-B* and original OPIM-C is that the size of all possible solutions is at most $\binom{n}{k_{max}} + n$ when $k_{max} \leq n/2$. Define $\mathcal{S}(B)$ as the set of all possible solutions with budget constraint B, it satisfies two properties: (i) The cost of a solution is at most B, and we cannot select any other nodes into the seed set, i.e., $\forall A \in \mathcal{S}(B)(A \subseteq V \wedge c(A) \leq B \wedge \forall v \in V \setminus A\,(c(A) + c(v) > B))$; (ii) Any two sets in $\mathcal{S}(B)$ do not intersect with each other, i.e., $\forall A, B \in \mathcal{S}(B)(A \neq B \Longrightarrow (A \nsubseteq B \wedge B \nsubseteq A))$. Based on the two properties, we can conclude that $|\mathcal{S}(B)| \leq \binom{n}{k_{max}}$ when $k_{max} \leq n/2$. This conclusion can be proven through a partial order diagram, and the relevant proof is omitted here. Finally, we add the number of single-node set solution n.

5.2.2 Bound the Approximation Ratio. We next derive the lower bound $\sigma^l(S_P)$ of $\sigma(S_P)$ and the upper bound $\sigma^u(S^o)$ of $\sigma(S^o)$, so as to bound the approximation ratio $\frac{\sigma(S_P)}{\sigma(S^o)} \geq \frac{\sigma^l(S_P)}{\sigma^u(S^o)}$. First, by the property of the greedy algorithm, we have

$$\mathrm{Cov}_{\mathcal{R}_1}(S_P) \geq (1 - 1/\sqrt{e})\mathrm{Cov}_{\mathcal{R}_1}(S^o). \tag{18}$$

Meanwhile, we set

$$\sigma^u(S^o) = \left(\sqrt{\frac{\mathrm{Cov}_{\mathcal{R}_1}(S_P)}{1 - 1/\sqrt{e}} + \frac{\ln(1/\delta_1)}{2}} + \sqrt{\frac{\ln(1/\delta_1)}{2}} \right)^2 \cdot \frac{n}{\Lambda_1}, \tag{19}$$

$$\sigma^l(S_P) = \left(\left(\sqrt{\mathrm{Cov}_{\mathcal{R}_2}(S_P) + \frac{2\ln(1/\delta_2)}{9}} - \sqrt{\frac{\ln(1/\delta_2)}{2}} \right)^2 - \frac{\ln(1/\delta_2)}{18} \right) \cdot \frac{n}{\Lambda_2}, \tag{20}$$

where $\Lambda_1 = |\mathcal{R}_1|$, $\Lambda_2 = |\mathcal{R}_2|$, and $\delta_1, \delta_2 \in (0, 1)$ are the probabilities of the worst cases. Then, we have the following lemma.

Lemma 5.1. *For any $\delta_1, \delta_2 \in (0, 1)$,*

$$\Pr[\sigma(S^o) \leq \sigma^u(S^o)] \geq 1 - \delta_1, \tag{21}$$

and

$$\Pr[\sigma(S_P) \geq \sigma^l(S_P)] \geq 1 - \delta_2. \tag{22}$$

Lemma 5.1 employs $\mathrm{Cov}_{\mathcal{R}_1}(S_P)/(1 - 1/\sqrt{e})$ as a basic upper bound of $\mathrm{Cov}_{\mathcal{R}_1}(S^o)$, which might be loose in practice. Therefore, we design a tightened upper bound of $\mathrm{Cov}_{\mathcal{R}_1}(S^o)$ as follows.

5.2.3 The Tightened Bound. Let the sequence of nodes $SQ(S) = \{v_1, v_2, v_3, \ldots, v_n\}$ be sorted by marginal coverage $\mathrm{Cov}_{\mathcal{R}_1}(v|S)$ of the unit cost. Denote that $SQ^u(S, B)$ is the set of maximum first k nodes in $SQ(S)$ that $\sum_{i=1}^{k} c(v_i) < B$, and $v_f(S, B)$ is the $k+1$ node after $SQ^u(S, B)$. We observe that an upper bound of $\mathrm{Cov}_{\mathcal{R}_1}(S^o)$ can be described in the following lemma.

Lemma 5.2. *For any seed set S_P and budget B, by setting*

$$Cov_{\mathcal{R}_1}(S_P, B) = Cov_{\mathcal{R}_1}(S_P) + \sum_{v \in SQ^u(S_P, B)} Cov_{\mathcal{R}_1}(v|S_P) + \frac{B - c(SQ^u(S_P, B))}{c(v_f(S_P, B))} Cov_{\mathcal{R}_1}(v_f(S_P, B)|S_P), \tag{23}$$

we have

$$Cov_{\mathcal{R}_1}(S^o) \leq Cov_{\mathcal{R}_1}(S_P, B). \tag{24}$$

Proof.

$$\begin{aligned}\mathrm{Cov}_{\mathcal{R}_1}(S^o) &\leq \mathrm{Cov}_{\mathcal{R}_1}(S_P) + \sum_{v \in S^o \setminus S_P} \mathrm{Cov}_{\mathcal{R}_1}(v|S_P) \\ &\leq \mathrm{Cov}_{\mathcal{R}_1}(S_P) + \sum_{v \in SQ^u(S,B)} \mathrm{Cov}_{\mathcal{R}_1}(v|S_P) + \frac{B - c(SQ^u(S,B))}{c(v_f(S,B))} \mathrm{Cov}_{\mathcal{R}_1}(v_f(S,B)|S_P),\end{aligned} \tag{25}$$

where the second inequality is by the definition of $SQ(S)$, that $\sum_{v \in S^o \setminus S_P} \mathrm{Cov}_{\mathcal{R}_1}(v|S_P)$ is no more than the marginal coverage of the most cost-effective nodes. □

Consequently, a tightened upper bound $\sigma^u(S^o)$ is

$$\sigma^u(S^o) = \left(\sqrt{\mathrm{Cov}^u_{\mathcal{R}_1}(S_P) + \frac{\ln(1/\delta_1)}{2}} + \sqrt{\frac{\ln(1/\delta_1)}{2}} \right)^2 \cdot \frac{n}{\Lambda_1}, \tag{26}$$

where

$$\mathrm{Cov}^u_{\mathcal{R}_1}(S_P) = \min \left\{ \frac{\mathrm{Cov}_{\mathcal{R}_1}(S_P)}{1 - 1/\sqrt{e}}, \min_{0 \leq i \leq |S_P|} \{\mathrm{Cov}_{\mathcal{R}_1}(S_i, B)\} \right\}, \tag{27}$$

and S_i is the seed set in the ith iteration.

According to Algorithm 5 and Lemma 5.1, setting $\delta_1 = \delta_2 = \delta/(3i_{max})$, we derive a desirable $\sigma(S^o) \leq \sigma^u(S^o)$ or $\sigma(S_P) \geq \sigma^l(S_P)$ with probability at least $1 - \delta/(3i_{max})$ in a given iteration i of i_{max} iterations. Therefore, $\frac{\sigma(S_P)}{\sigma(S^o)} \geq \frac{\sigma^l(S_P)}{\sigma^u(S^o)}$ with probability at least $1 - 2\delta/(3i_{max})$. Moreover, the failure probability for the result after the final iteration is at most $\delta/3$. By the union bound, the algorithm OPIM-B yields a $1 - 1/\sqrt{e} - \epsilon$, with probability at least $1 - \delta$.

5.3 SBG: Sandwich Approximation Framework for the BCIM Problem on the GCLT Model

In this section, we apply the Sandwich Approximation framework [28] to design the SBG algorithm for the BCIM problem on the GCLT model. The main idea is to solve optimization problems on the lower and upper bound functions to obtain a solution that can be bounded. First, we find an approximate solution to the lower bound and the upper bound by OPIM-B algorithm. Then, we select the seed set of solutions that achieves the best influence spread in Monte Carlo simulations.

5.3.1 Preliminaries. We review the (ϵ, δ)-approximation [12], which will be used in the algorithm.

Definition 3 ((ϵ, δ)-Approximation). Let $Z_1, Z_2, \ldots$ be samples distributed independently and identically according to Z with mean μ_Z. A Monte Carlo estimator of μ_Z,

$$\mu_Z = \frac{1}{T} \sum_{i=1}^{T} Z_i, \tag{28}$$

is said to be an (ϵ, δ)-approximation of μ_Z if

$$\Pr[(1 - \epsilon)\mu_Z \leq \hat{\mu}_Z \leq (1 + \epsilon)\mu_Z] \geq 1 - \delta. \tag{29}$$

To achieve that approximation, Lemma 5.3 shows a criterion.

Algorithm 6: Sandwich Approximation for the BCIM Problem on the GCLT Model (SBG)

Input: Graph $G(V, E, P, w, c)$, negative seed set S_N, budget B and $\epsilon, \epsilon', \delta, \delta' \in (0, 1)$
Output: Seed set S_P

Let S_L be the output seed set of Algorithm OPIM-B for the Lower bound.
Let S_U be the output seed set of Algorithm OPIM-B for the upper bound.
for S_L and S_U **do**
  Let $\mu_Z = I(S)/n$, estimate $\hat{\mu}_Z$ using Lemma 5.4 with $\Upsilon_1 = 1 + (1 + \epsilon')\Upsilon(\delta'/8, 1/2)$.
  Set $N = 2\Upsilon(\delta'/4, \epsilon') \cdot \epsilon'/\hat{\mu}_Z$ and X_i be the influence spread percentage of the i-th simulation.
  $\hat{\sigma}_Z^2 \leftarrow \frac{1}{N}\sum_{i=1}^{N}(X_{2i-1} - X_{2i})^2/2$.
  $\hat{\rho}_Z \leftarrow max\{2\hat{\sigma}_Z^2, 2\epsilon'\hat{\mu}_Z\}$
  Set $N = 9\Upsilon(\delta'/4, \epsilon') \cdot \hat{\rho}_Z/4\hat{\mu}_Z^2$ and Y_i be the influence spread of the i-th simulation.
  $\hat{I}(S) \leftarrow \frac{n}{N}\sum_{i=1}^{N} Y_i$
$S_P \leftarrow \arg\max_{S' \in \{S_L, S_U\}} \hat{I}(S')$
return S_P

Lemma 5.3. *Let $Z_1, Z_2, \ldots, Z_N$ be samples distributed independently and identically according to Z in the interval $[0,1]$ with mean μ_Z and variance σ_Z^2. Let $Sum_Z = \sum_{i=1}^{N} Z_i$ and $\hat{\mu}_Z = Sum_Z/N$. If $N(\delta/2, \beta) = 4(e-2)\ln(2/\delta)\rho_Z/\beta^2$, $\rho_Z \geq \sigma_Z^2$, and $\beta \leq 2(e-2)\rho_Z$, then*

$$\Pr[\hat{\mu}_Z - \mu_Z \geq \beta] \leq \delta, \tag{30}$$

and

$$\Pr[\hat{\mu}_Z - \mu_Z \leq -\beta] \leq \delta. \tag{31}$$

If only one of Equations (30) and (31) is required, then the number of samples N becomes

$$N(\delta, \beta) = 4(e-2)\ln\frac{1}{\delta}\frac{\rho_Z}{\beta^2}. \tag{32}$$

Let $\rho_Z = \mu_Z$ and $\beta = \epsilon\mu_Z$, $\Upsilon(\delta/2, \epsilon) = 4(e-2)\ln(2/\delta)/\epsilon^2$, we have the following lemma.

Lemma 5.4. *Let $\Upsilon_1 = 1 + (1+\epsilon)\Upsilon(\delta/2, \epsilon)$ and $\hat{\mu}_Z = \Upsilon_1/N$. If N is the number of samples when $Sum_N \geq \Upsilon_1$ and $\epsilon < 1$, then $\mathbb{E}[N] \leq \Upsilon_1/\mu_Z$ and*

$$\Pr[(1-\epsilon)\mu_Z \leq \hat{\mu}_Z \leq (1+\epsilon)\mu_Z] \geq 1 - \delta. \tag{33}$$

5.3.2 Sandwich Approximation. The solution to the BCIM problem in the GCLT model is shown in Algorithm 6. In the algorithm, we give an estimation of $I(S)$ with a theoretical guarantee, which can be proved by the following lemma.

Lemma 5.5. *The estimation of $I(S)$ in Algorithm 6 is (ϵ', δ')-approximation, that is*

$$\Pr[(1-\epsilon')I(S) \leq \hat{I}(S) \leq (1+\epsilon')I(S)] \geq 1 - \delta'. \tag{34}$$

Proof. By Lemma 5.4, we estimate μ_Z that

$$\Pr\left[\frac{1}{2}\mu_Z \leq \hat{\mu}_Z \leq \frac{3}{2}\mu_Z\right] \geq 1 - \frac{\delta}{4}. \tag{35}$$

Then, we estimate σ_Z^2 with $N = 2\Upsilon(\delta'/4, \epsilon') \cdot \epsilon'/\hat{\mu}_Z$ simulations, and set $\hat{\rho}_Z = max\{2\hat{\sigma}_Z^2, 2\epsilon'\hat{\mu}_Z\}$.

(1) If $\sigma_Z^2 < 2\epsilon'\hat{\mu}_Z$, then $\hat{\rho}_Z \geq 2\epsilon'\hat{\mu}_Z > \sigma_Z^2$ holds.
(2) If $\sigma_Z^2 \geq 2\epsilon'\hat{\mu}_Z$, then $\frac{1}{2}\sigma_Z^2 \leq \hat{\sigma}_Z^2$ holds with probability at least $1 - \delta'/4$.

Let $Var(\sigma_Z^2)$ be the variance of σ_Z^2, we have

$$Var(\sigma_Z^2) = E[(X-\mu_Z)^4] - \sigma_Z^4 \le E[(X-\mu_Z)^2] - \sigma_Z^4 \le \sigma_Z^2. \tag{36}$$

To achieve this inequality $\frac{1}{2}\sigma_Z^2 \le \hat{\sigma}_Z^2$ by Lemma 5.3, the number of simulations is at least

$$\begin{aligned} 4(e-2)\ln(4/\delta')\sigma_Z^2 / \left(\frac{\sigma_Z^2}{2}\right)^2 &= 8(e-2)\ln(4/\delta') / \left(\frac{\sigma_Z^2}{2}\right) \\ &\le 8(e-2)\ln(4/\delta')/\epsilon'\hat{\mu}_Z \\ &= 2\Upsilon(\delta'/4,\epsilon')\cdot\epsilon'/\hat{\mu}_Z. \end{aligned} \tag{37}$$

Then, there are $\hat{\rho}_Z \ge 2\hat{\sigma}_Z^2 \ge \sigma_Z^2$ and $\hat{\rho}_Z \ge 2\epsilon'\hat{\mu}_Z \ge \epsilon'\mu_Z$.

Therefore, by condition $\hat{\mu}_Z \le \frac{3}{2}\mu_Z$ and setting $N = 9\Upsilon(\delta'/4,\epsilon')\cdot\hat{\rho}_Z/4\hat{\mu}_Z^2$, we have

$$\Pr[(1-\epsilon')I(S) \le \hat{I}(S) \le (1+\epsilon')I(S)] \ge 1 - \frac{\delta'}{2}, \tag{38}$$

with probability at least $1-\delta'/2$. By the union bound, the lemma is proved. □

We can prove that the following theoretical result exists for the sandwich approximation framework.

Theorem 5.1. *Assume S_P is the seed set obtained from Algorithm 6. We have*

$$I(S_P) \ge \max\left\{\frac{I(S_U)}{U(S_U)}, \frac{L(S_L^*)}{I(S_P^*)}\right\}\frac{1-\epsilon'}{1+\epsilon'}\left(1, \frac{1}{\sqrt{e}} - \epsilon\right) I(S_P^*), \tag{39}$$

where S_L^* is the optimal solution to the problem of maximum budgeted coverage in the lower bound, and S_P^* is the optimal solution to the original BCIM problem under the GCLT model.

Proof. Assume S_U^* is the optimal solution to solve the problem of maximum budgeted coverage in the upper bound. Algorithm 5 guarantees a $(1-1/\sqrt{e}-\epsilon)$-approximation, then, we have

$$I(S_U) = \frac{I(S_U)}{U(S_U)}U(S_U) \ge \frac{I(S_U)}{U(S_U)}\left(1 - \frac{1}{\sqrt{e}} - \epsilon\right)U(S_U^*). \tag{40}$$

Due to $U(S_U^*) \ge U(S_P^*)$ and $U(S_P^*) \ge I(S_P^*)$ since $U(\cdot)$ is an upper bound of $I(\cdot)$. Then

$$I(S_U) \ge \frac{I(S_U)}{U(S_U)}\left(1 - \frac{1}{\sqrt{e}} - \epsilon\right)U(S_P^*) \ge \frac{I(S_U)}{U(S_U)}\left(1 - \frac{1}{\sqrt{e}} - \epsilon\right)I(S_P^*). \tag{41}$$

On the other hand,

$$I(S_L) \ge L(S_L) \ge \left(1 - \frac{1}{\sqrt{e}} - \epsilon\right)L(S_L^*) \ge \frac{L(S_L^*)}{I(S_P^*)}\left(1 - \frac{1}{\sqrt{e}} - \epsilon\right)I(S_P^*). \tag{42}$$

Let S_{max} be the solution to Algorithm 6 and S_{min} be with the minimum $\hat{I}(S_{min})$, the estimation method is (ϵ', δ)-approximation. We have

$$(1-\epsilon')I(S_{max}) \le \hat{I}(S_{max}) \le (1+\epsilon')I(S_{max}) \tag{43}$$

and

$$(1-\epsilon')I(S_{min}) \le \hat{I}(S_{min}) \le \hat{I}(S_{max}). \tag{44}$$

It follows that

$$I(S_P) \ge \frac{1}{1+\epsilon'}\hat{I}(S_{max}) \ge \max\left\{\frac{I(S_U)}{U(S_U)}, \frac{L(S_L^*)}{I(S_P^*)}\right\}\frac{1-\epsilon'}{1+\epsilon'}\left(1 - \frac{1}{\sqrt{e}} - \epsilon\right)I(S_P^*). \tag{45}$$

□

Algorithm 7: RISIEA Algorithm

Input: Graph $G(V, E, P, w, c)$, negative seed set S_N, positive seed set S_P and $\epsilon', \delta' \in (0, 1)$
Output: Expected Influence $\hat{I}(S_P)$

1: $\Upsilon' \leftarrow 1 + 4(e-2)(1+\epsilon')\ln(2/\delta')/\epsilon'^2$.
2: $N \leftarrow 0$, $a \leftarrow 0$
3: **while** $a < \Upsilon'$ **do**
4: $N \leftarrow N + 1$
5: Randomly select a node $v \in V$
6: Generate a URR set R_U of node v by RIS
7: **if** $\mathrm{Cov}_{R_U}(S_P) = 0$ **then continue**
8: $x \leftarrow GN(v)$
9: Reverse reachable node set $R \leftarrow \{x\}$, active node set $Q \leftarrow \emptyset$
10: **while** $R \neq \emptyset$ and $x \notin Q$ **do**
11: $R \leftarrow reverse(R)$ ▷ Search reverse reachable nodes using BFS.
12: $Q \leftarrow Q \cup (R \cap S_N) \cup (R \cap S_P)$
13: $Q \leftarrow forward(Q)$ ▷ Spread forward and update.
14: **if** $x \in Q$ and x is positive activated **then** $a \leftarrow a + 1$
15: **if** $x \notin Q$ **then** $R \leftarrow \{v\}$ and repeat the while loop
16: **return** $\hat{I}(S_P) \leftarrow n * \Upsilon'/N$

5.3.3 Discussion on the Tightness of the Bounds. The proposed SBG model consists of two parts. The upper and lower bounds only affect the selection of the seed set in the first part. A tighter bound usually leads to a better seed set with higher quality, which indicates a wider influence spread. However, due to the characteristics of the model, this will make it more difficult to calculate the reverse reachable set, resulting in lower time efficiency. Moreover, compared to the upper and lower bounds given in the article, a tighter bound mainly considers some marginal cases, which usually have less significance in actual social network environments.

5.4 Possible Optimizations

In this section, we discuss the possible optimizations of our *SBG* model.

5.4.1 SBG-R: Improving Influence Estimation Based on RIS. The estimation of the influence spread of the candidate seed sets in *SBG* is very time-consuming. Therefore, we further propose the **reverse influence sampling-based influence estimation algorithm (RISIEA)** (in Algorithm 7). Replacing the influence estimation part of Algorithm 6 with Algorithm 7, we obtain the optimized model *SBG-R*.

For the GCLT model, we cannot quickly estimate the influence spread of any seed set by constructing the reverse reachable set and calculating the set coverage as in the LT model. However, when the seed sets are determined, the influence spread can still be estimated by calculating the probability that the random node will be activated.

Algorithm 7 initially defines the constant Υ' (Line 1) as the stopping condition following Lemma 5.4. When the number a of positive activated random nodes v reaches Υ' (Line 3), the algorithm stops and returns the estimate of influence spread $\hat{I}(S_P) = n * \Upsilon'/N$ (Line 16), where N is the number of simulations. For each simulation, it first randomly selects a node $v \in V$ and then generates a URR set R_U of node v by RIS (Lines 5 and 6). If the URR set R_U is not covered by the seed set S_P, then it is impossible to positively activate node v (Line 7). If the URR set R_U is covered, it simulates whether the group node $GN(v)$ of v will be activated by the seed sets (Lines 8–13). If $GN(v)$ is positively activated, then the number a increases and continues (Line 14). If $GN(v)$ is not activated, then it simulates whether node v will be activated (Lines 15 and 10–13). In each simulation, it starts from a root node v. R is the set of reverse reachable nodes in each iteration,

which is updated by the procedure *reverse(R)*. In the procedure *reverse(R)*, it searches the reverse reachable nodes and their group nodes in the positive and negative GCLE sample graphs using the BFS method. Q is the set of active nodes. If Q is updated by the set R, it spreads forward through the sampled GCLE graphs (*forward(Q)*) until it reaches the root node.

5.4.2 SBG-M: Improving the Time Efficiency of Monte Carlo Simulation. We further propose a variant of *SBG-M* to improve the time efficiency. *SBG* estimates $I(S_U)$ and $I(S_L)$ using a Monte Carlo simulation. This step is also very time-consuming. In a Monte Carlo simulation, it samples a graph using the GCLE model and estimates the spread, and the estimation process of the two sets is uncorrelated. To improve the efficiency, we can estimate $I(S_U)$ and $I(S_L)$ in the same sample graph for each simulation.

In a sample GCLE graph, a node can be influenced by at most one node. After a simulation of a seed set, we can record which nodes can be influenced by the nodes that have been passed through. Then, in the next simulation, we can ignore the edges that will never be spread by. Meanwhile, the competitive node set is the same in each simulation, and the nodes they pass through will also be quite similar. Using this method, the overall Monte Carlo simulation time can be reduced. Note that *SBG-M* mainly improves the simulation time, while the time complexity of *SBG* remains.

We can also combine the above two optimizations and obtain the *SBG-RM* model.

5.5 Complexity Analysis

In this section, we analyze the time and space complexities of the proposed model.

5.5.1 Time Complexity. Algorithms 1, 2, 5, and 6 together constitute the SBG procedure. Meanwhile, Algorithms 1, 2, 5, and 7 constitute the SBG-R procedure. And SBG-M has the same complexity as SBG.

Algorithms 1 and 2 are based on the RIS process. Although the algorithms consider group nodes, the complexity is limited to the length of the RR set. Therefore, the time complexity of Algorithms 1 and 2 is $O(|R|)$.

Let n and m be the number of nodes and edges, respectively. Then in Algorithm 5, it executes $i_{max} = \left\lceil \log_2 \frac{n}{\epsilon^2 I_1^g(S)} \right\rceil$ iterations. For each iteration, it takes $O(k_{max}n + \sum_{R\in\mathcal{R}_1} |R|)$ time to compute S_P and $\sigma^u(S^o)$ from $\mathcal{R}_1$. Recall that computing $\sigma^l(S_P)$ takes $O(\sum_{R\in\mathcal{R}_2} |R|)$ time. Therefore, the total time complexity of each iteration is $O(k_{max}n + \sum_{R\in\mathcal{R}_1\cup\mathcal{R}_2} |R|)$. Similar to algorithm OPIM-C [42], OPIM-B generates an expected number of $O((k_{max} \ln n + ln(1/\delta))n\epsilon^{-2}/\sigma(S^o))$ RR sets and runs in expected time $O((k_{max} \ln n + ln(1/\delta))(n+m)\epsilon^{-2})$ when budget $B \geq \max_{v\in V} c(v)$.

Algorithm 6 runs a Monte Carlo estimation of the seed sets. The expected number of simulations [12] is $E[N] \leq 4c'(e-2)\ln(2/\delta')\rho_Z/\epsilon^2\mu_Z^2$, where $c' > 1$ is a constant. Recall that $\rho_Z = max\{\sigma_Z^2, \epsilon\mu_Z\}$, and $\sigma_Z^2 \leq \mu(1-\mu)$ when $\mu \in [0,1]$, we have $\rho_Z < \mu_Z$ and $E[N] \leq 4c'(e-2)\ln(2/\delta')/\epsilon^2\mu_Z$. For a simulation, the time complexity can be regarded as $I(S_P)m/n + I(S_P)$, which is based on the influence spread $I(S_P)$ and the average degree m/n. Since $\mu_Z = I(S_P)/n$, the time complexity is

$$\begin{aligned} O(E[N]\cdot(I(S_P)m/n + I(S_P))) &= O((I(S_P)m/n + I(S_P))\ln(2/\delta')/\epsilon^2\mu_Z) \\ &= O((I(S_P)m/n + I(S_P))\ln(2/\delta')n/\epsilon^2 I(S_P)) \\ &= O(\ln(2/\delta')(m+n)\epsilon^{-2}). \end{aligned} \tag{46}$$

Meanwhile, the time complexity of the spread process of S_N is $O(\ln(2/\delta')(m+n)\epsilon^{-2}\frac{I(S_N)}{I(S_P)})$. Therefore, the time complexity of Monte Carlo estimation in Algorithm 6 is $O(\ln(2/\delta')(m+n)\epsilon^{-2}(\frac{I(S_N)}{I(S_P)} + 1))$.

Table 2. Dataset Statistics

Dataset	#Nodes	#Edges	#Groups	Max Group Size
NetHEPT [10]	15K	59K	1,244	76
NetPHY [10]	37K	181K	3,070	203
Amazon [53]	334K	925K	8,075	262
YouTube [53]	1.1M	3M	12,875	7,185
LiveJournal [53]	4M	34.7M	15,729	8,249
Orkut [53]	3M	117M	12,164	7,680

In Algorithm 7, the expected number of simulations is $E[N] \leq (1+4(e-2)(1+\epsilon')\ln(2/\delta')/\epsilon'^2)/\mu_Z$. The time complexity of one simulation is $O(n)$. Then the time complexity is $O(\ln(2/\delta')n^2/\epsilon'^2 I(S_P))$.

In summary, the total time complexity of SBG/SBG-M is $O((k_{max}\ln n + ln(1/\delta))(n+m)\epsilon^{-2}) + O(\ln(2/\delta')(m+n)\epsilon^{-2}(\frac{I(S_N)}{I(S_P)}+1))$ and that of SBG-R is $O((k_{max}\ln n + ln(1/\delta))(n+m)\epsilon^{-2}) + O(\ln(2/\delta')n^2/\epsilon'^2 I(S_P))$.

5.5.2 Space Complexity. The space consumption is $O(m+n+\sum_{R\in\mathcal{R}_1\cup\mathcal{R}_2}|R|)$ consisting of the graph G and the RR sets. Previous work [44] shows that the expected length of a random RR set is $E[\sigma(\{v*\})]$ where $v*$ is a node selected randomly from those in G with probabilities proportional to their in-degrees. Because $\sigma(S^o) \geq \sigma(\{v*\})$ when $B \geq \max_{v\in V} c(v)$, the space complexity is $O(m+n+(k_{max}\ln n + ln(1/\delta))n\epsilon^{-2})$.

6 Experiments

In this section, we compare our SBG algorithm with six baselines on six real datasets, with respect to the influence spread, the running time, the number and runtime of Monte Carlo simulations, the effects of the acceptability parameter, and experiments of group consensus. All the experiments were conducted on a server with a 2.1 GHz Intel Xeon 8 Core CPU and 128 GB memory running CentOS/7.2 OS. All the code is written in Java.[1]

6.1 Experimental Setup

Datasets: We use six representative real-world network datasets of different scales from thousands to millions of nodes and edges: NetHEPT, NetPHY, Amazon, YouTube, LiveJournal, and Orkut. NetHEPT and NetPHY are citation networks of physics papers, Amazon is the product co-purchasing network, YouTube and Orkut are OSNs, and LiveJournal is a free online blogging community. The statistics of those datasets are shown in Table 2, in which 1 K represents 1,000 and 1 M means 1 million, respectively. In this article, the proposed GCLT model is based on a network with non-overlapping group structures. Therefore, in our experiments, we do not use the original communities in the datasets. Instead of that, we divide the social networks into small, tightly connected groups, using the improved Louvain algorithm [46]. *# Groups* represents the total number of groups with more than three nodes in the datasets. We discussed how our work can be extended to dealing with overlapping groups in Section 3.1, and we will leave the detailed implementation of GCLT in overlapping group structures for future work.

Figure 6 shows the distribution of group sizes in the six datasets. The group size of the NetHEPT and NetPHY datasets is generally smaller than that of the other four larger datasets. Meanwhile, most groups are not large in all the datasets.

[1]The code can be found at: https://github.com/csjwj2023/Group-Consensus-based-Competitive-Influence-Maximization.

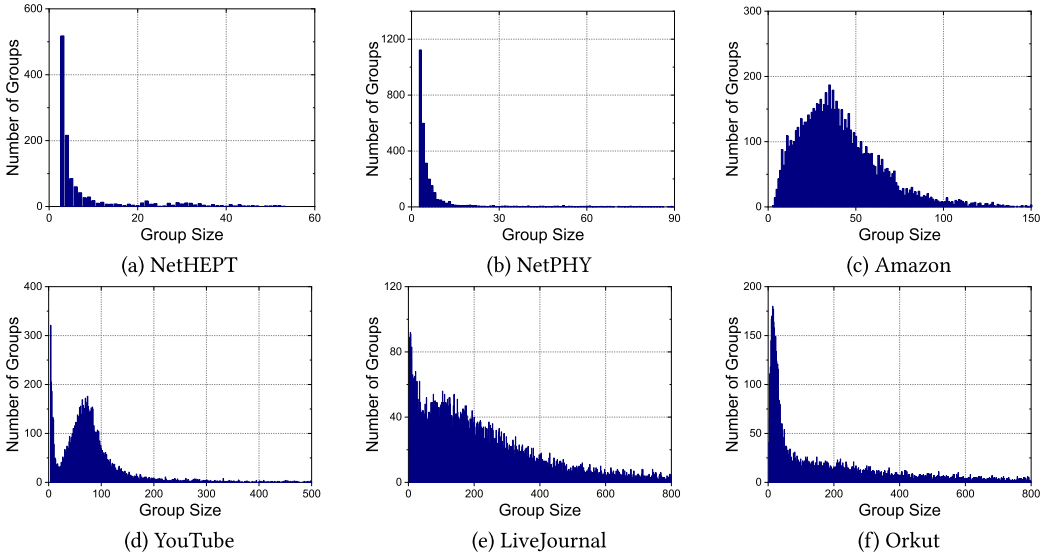


Fig. 6. The distribution of group size.

Comparison Algorithms: In the experiments, we will compare our SBG with representative baselines. Since our work studies group-based IM and CIM with budget constraints and our model provides approximation theoretical guarantees of the bounds of the objective function, we select and implement three types of related and representative methods and a total of six baselines, as follows:

— *Group-Based Methods*: (1) The **weighted degree (WD)** heuristic method, in which each node is taken as a group. The nodes with the Top-k largest weighted sums of outgoing degrees are selected as the seed node, and the nodes in the competitor seed set are skipped. (2) The group-based degree heuristic algorithm *ComBIM* [1], which allocates the budget proportionally among different groups according to the proportion of nodes from different groups to the total number of nodes and exploits a maximum degree heuristic algorithm to select seed nodes within each group. (3) The **group consensus-based degree discount (GDD)** heuristic algorithm, which is designed by extending the degree heuristic [10] for the GCLT model. It calculates the gain of selecting each node by Equation (47) and iteratively adds a node v with the maximum gain over cost ratio $E(v)/c(v)$ and removes v from the candidate set U. If the cost of node v is less than the remaining budget, add it to the seed set S_P, and update the gain of each in-neighbor of v and the gain of each node in the same group as v using Equation (47):

$$E(v) = n(C \setminus S_P) \cdot w(v, C) + \sum_{u \in out(v) \setminus (S_N \cup S_P)} w^+(v, u). \tag{47}$$

Equation (47) divides the one-step influence of node v into two parts: the expected influence gained by activating group C, where $n(C \setminus S_P)$ represents the number of all nodes in group C excluding the nodes in the positive seed set S_P, and $w(v, C) = 1$ if $v \in C$, otherwise $w(v, C) = 0$; and the expected influenced outgoing nodes of v, which excludes the nodes in the competitor seed set S_N and the positive seed set S_P.

— *RIS-Based Methods*: (1) *DSSA-B* that we specially modified for the BIM problem based on DSSA [35], considering the budget constraints. (2) *OPIM-B+* for the BIM problem, which is extended

from OPIM-C [42], using the improved upper bound, i.e., Equation (26) in our work. *OPIM-B+* selects the reverse reachable set with the basic RIS algorithm [4], which does not consider the influence of group consensus and competition and does not choose competing nodes to the seed set, either. Both *DSSA-B* and *OPIM-B+* can provide approximation theoretical guarantees of the bounds of the objective function.

— *Sandwich Approximation Framework-Based Method*: SPBA [39], an approximation algorithm based on the Sandwich framework and polling-based method, is designed for the BCIM problem, which considers budgets and competitive nodes, but neglects the group consensus. SPBA can also provide approximation theoretical guarantees of the bounds of the objective function.

It is worth noting that several new studies have been done, e.g., [8, 50, 51, 56]. But they are not suitable for comparison because they define new, different problems for IM.

Parameter Settings: In all experiments, we keep $\epsilon = \epsilon' = 0.1$, $\delta' = 0.01$, and $\delta = 1/n$ as general settings. The weight of edge (u, v) is calculated as $w(u, v) = \frac{1}{d_{in}(v)}$ where $d_{in}(v)$ denotes the in-degree of node v. The weight of node v to the group C is calculated as $w(v, C) = \frac{d_{out}(v)+1}{\sum_{u \in C}(d_{out}(v)+1)}$. We set the strength of group consensus λ by Equation (2) for default. The influence spread evaluation is obtained with 10,000 Monte Carlo simulations.

Intuitively, the more famous one is, the more difficult it is to convince that person. Hence, we assign the cost of a node positively correlated to its out-degree: $c(u) = d_{out}(v) + 1$ where $d_{out}(v)$ is the out-degree of node v.

6.2 Experimental Results

6.2.1 Comparison of Influence Spread. In this experiment, we compare the influence spread of the algorithms with varied budgets B, according to the size of the networks in different datasets. The results are displayed in Figure 7. Generally, the influence spread of our SBG method outperforms other methods on the BCIM problem under the GCLT model in all six datasets, validating the effectiveness of our work. We have the following main findings.

The influence spread of SBG is larger than other methods, and SPBA or GDD lies in the second. The improvements are more significant in the four large datasets, i.e., Amazon, YouTube, LiveJournal, and Orkut. To be specific, the influence spread of SBG is 6.13–18.85% larger than that of the second-best baseline SPBA on Amazon, 9.12–28.75% larger on YouTube, 10.39–79.33% larger on LiveJournal, and 13.17–59.53% larger on Orkut. We analyze the reason and find that this is because the methods except SBG and GDD ignore the group consensus in the influence diffusion process and neglect the spread of group activation. Moreover, GDD performs better in five datasets except Orkut, indicating its advantages in IM, but also reflecting its drawback of not being stable. This further reflects the advantages of our SBG, which exploits the sandwich framework and performs stably.

Generally, the influence spreads of ComBIM, DSSA-B, and OPIM-B+ lie in the middle, and WD is the last in most datasets. The four baselines, particularly WD, have much worse performance on influence spread. The reason may be that they select expensive nodes that cost a lot, or ignore the competitive influence. While the seed set returned by our SBG algorithm considering both cost and competitive influence, thus has a higher influence. Moreover, SBG and SPBA take advantage of the upper and lower bounds of the objective function to obtain the approximation ratio.

6.2.2 Comparison of Running Time. We compare the running time of our SBG based on the upper and lower bounds of the objective function (denoted as SBG-UR and SBG-LR, respectively), with five baselines except WD, which has too low influence spread: GDD, ComBIM, DSSA-B, OPIM-B+, and SPBA based on the lower bound of the objective function designed for the CLT

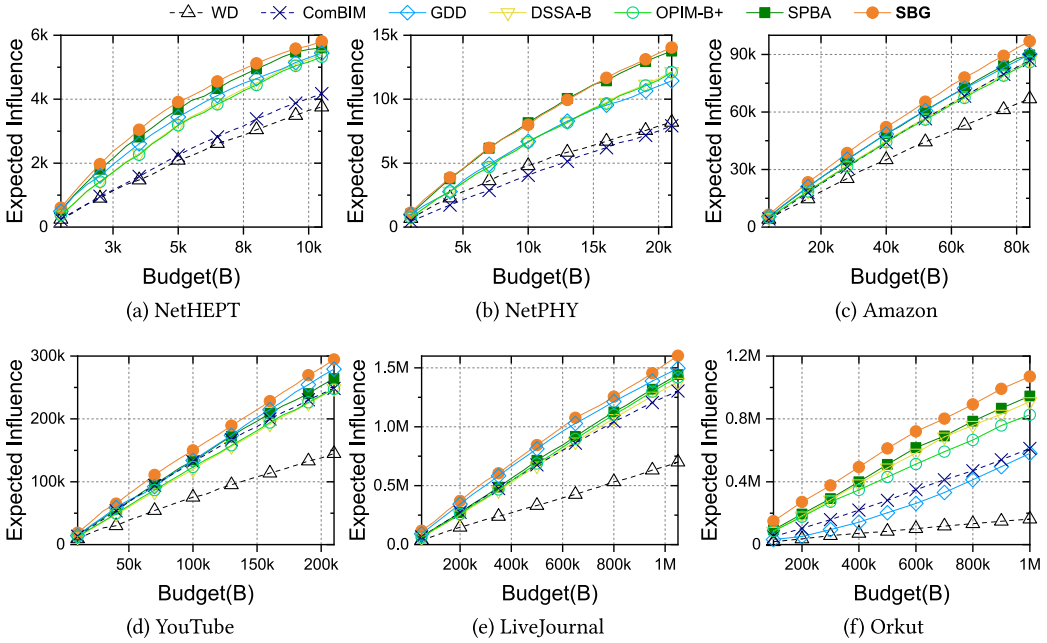


Fig. 7. Comparison of influence spread between different methods.

model (denoted as SPBA-LR). The node cost is set to 1. The results are shown in Figure 8. We have the following main findings.

GDD, ComBIM, and OPIM-B+ have a low running time in most datasets. Moreover, the influence spread of GDD is very close to that of SBG. Although ComBIM has a low running time, its influence spreads are also smaller, as shown in Figure 7. Therefore, the group consensus-based GDD algorithm maintains a good balance between influence spread and running time, indicating the advantage of considering group consensus. Meanwhile, OPIM-B+ has a low running time; the reason may be that it improves the efficiency by our improved upper bound in Equation (26).

In most cases, the running time of SPBA-LR lies between that of our SBG-UR and SBG-LR algorithms. Moreover, our algorithms SBG-UR and SBG-LR also maintain low running times on large datasets, e.g., less than 10 seconds on YouTube, less than 100 seconds on LiveJournal, and less than 200 seconds on Orkut. Meanwhile, DSSA-B has the longest running time on most datasets. We analyze the reason and find that the sampling process is more complex in DSSA-B.

6.2.3 Comparison of Simulation Runtime and Number of Simulations. We compare the simulation runtime and number of simulations of our SBG algorithm with SPBA, which also exploits Monte Carlo simulations. We also implemented three variants of our SBG algorithm, SBG-R, SBG-M, and SBG-RM, which improve the time efficiency of Monte Carlo simulations (Section 5.4). The results are shown in Figures 9 and 10. Note that SBG-M does not change the number of simulations, but improves the efficiency of each simulation. Therefore, in Figure 10, we do not display the results of SBG-M and SBG-RM, which are the same as SBG and SBG-R, respectively. We have the following main findings.

The simulation runtime and number of simulations decrease as the budget increases. With the increase of the budget, which means more seed nodes can be selected, both the simulation runtime and the number of simulations of all the methods decrease significantly. To be specific, the simulation

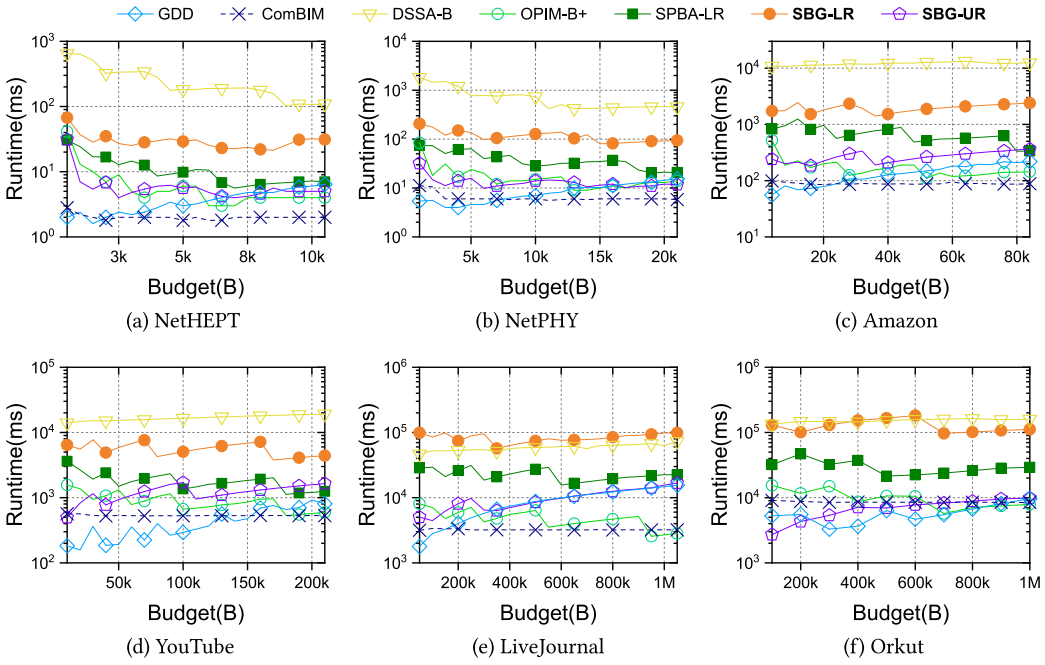


Fig. 8. Comparison of running time of different methods.

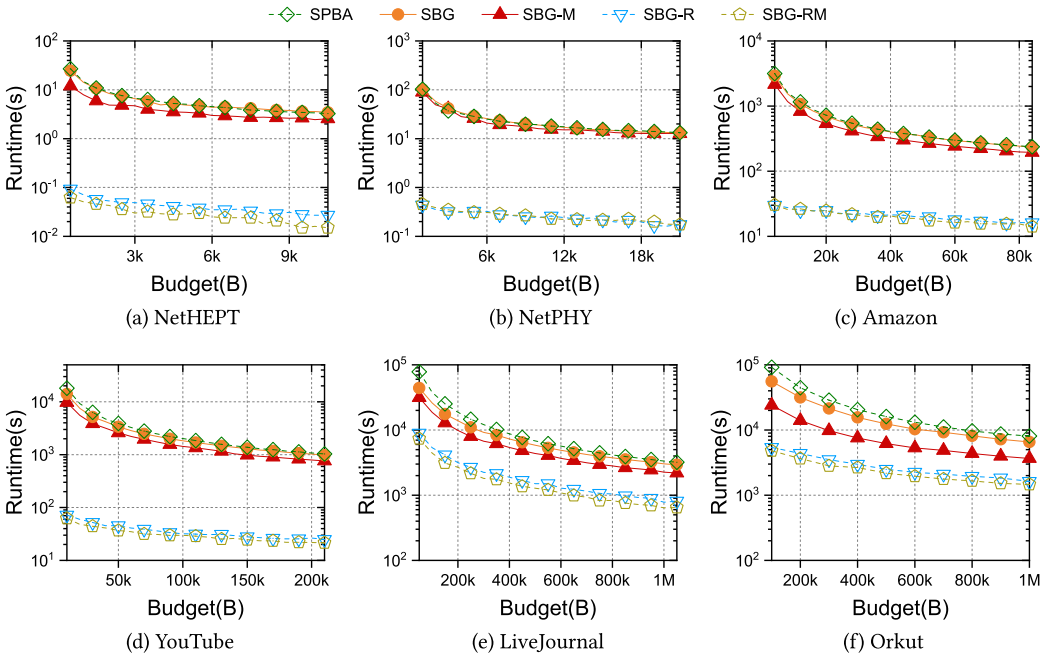


Fig. 9. Comparison of simulation runtime of different methods.

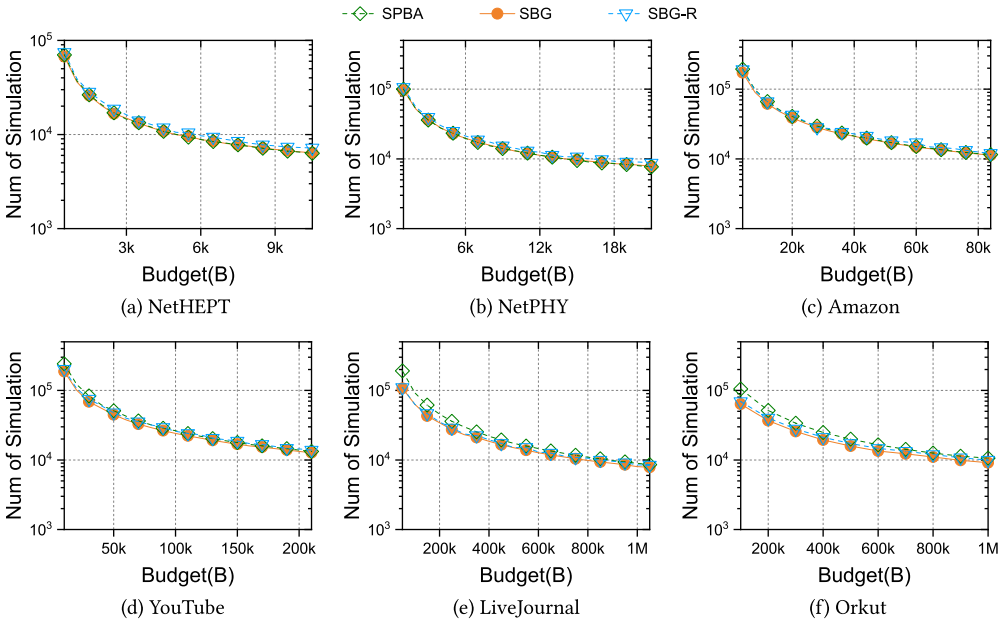


Fig. 10. Number of simulations of different methods.

runtime of the largest budget remains 4.11–53.12% of that of the smallest budget; and the number of simulations of the largest budget remains 4.48–14.36% of that of the smallest budget. Taking SBG-RM on LiveJournal as an example, the simulation runtime is 7,340.114 ms when the budget is 50,000 while it becomes 8.84%, i.e., 648.604 ms when the budget is 1,050,000; and the number of simulations is 110,112 versus 8,456, remaining 7.68%. It indicates that more seed nodes help to promote the influence to spread faster.

Our SBG algorithm has a lower simulation runtime and fewer number of simulations in all six datasets. It shows that our SBG algorithm is more efficient than SPBA. The reason may be that our model integrates the group consensus, which boosts the activation of nodes in a group.

SBG-R and SBG-RM improve the simulation runtime significantly in all datasets. It shows that the simulation runtime of our SBG-R algorithm is 74.43–99.45% less than that of SPBA in all datasets, and the improvement of SBG-RM is 77.33–99.75%. This validates the effectiveness of our optimization strategies.

6.2.4 The Effect of Acceptability Parameter λ in Influence Spread. In real life, there are different requirements for group consensus for different scenarios or activities. The acceptability parameter λ determines how easily the nodes accept the group opinion. To study how the acceptability of group consensus can affect the influence diffusion process, we compare our SBG with GDD and SPBA algorithms, while varying λ from 0.2 to 1.0. We check two influence spreads with different λ, that is, the overall spread (Figure 11) and the partial of spread activated by groups (Figure 12).

With the increase of the group consensus acceptability, the overall influence spread of our SBG performs much better. Figure 11 shows the overall influence spreads when the budgets are 4,000 (NetHEPT), 10,000 (NetPHY), 20,000 (Amazon), 40,000 (YouTube), 100,000 (LiveJournal), and 200,000 (Orkut), respectively. Our SBG outperforms the other two algorithms. And as λ increases, SBG performs much better than SPBA. For example, compared to the overall spread of SPBA in Amazon, SBG improves by 1.61% when $\lambda = 0.2$ and by 11.56% when $\lambda = 1$; while the improvements in

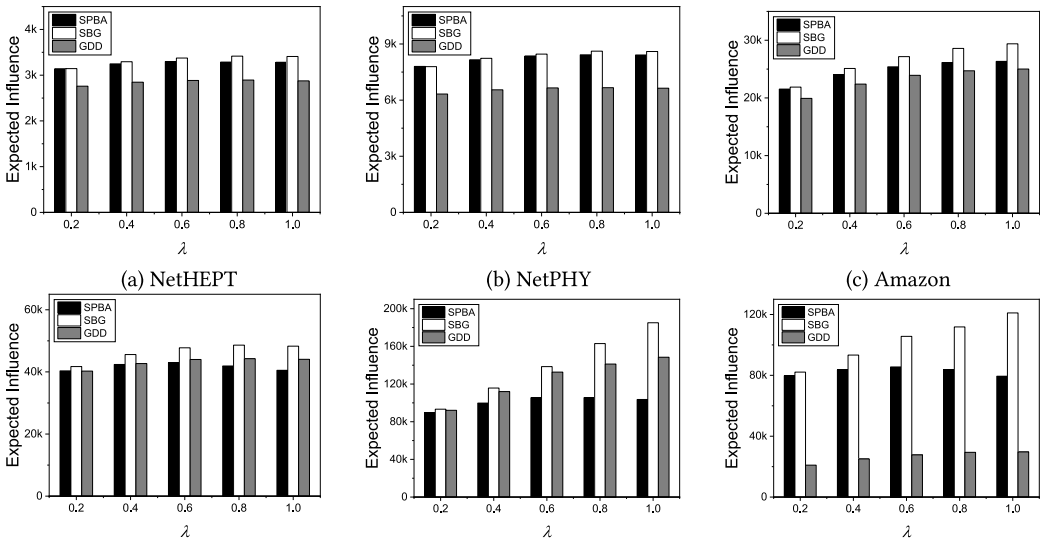


Fig. 11. The overall influence spread with different acceptability λ.

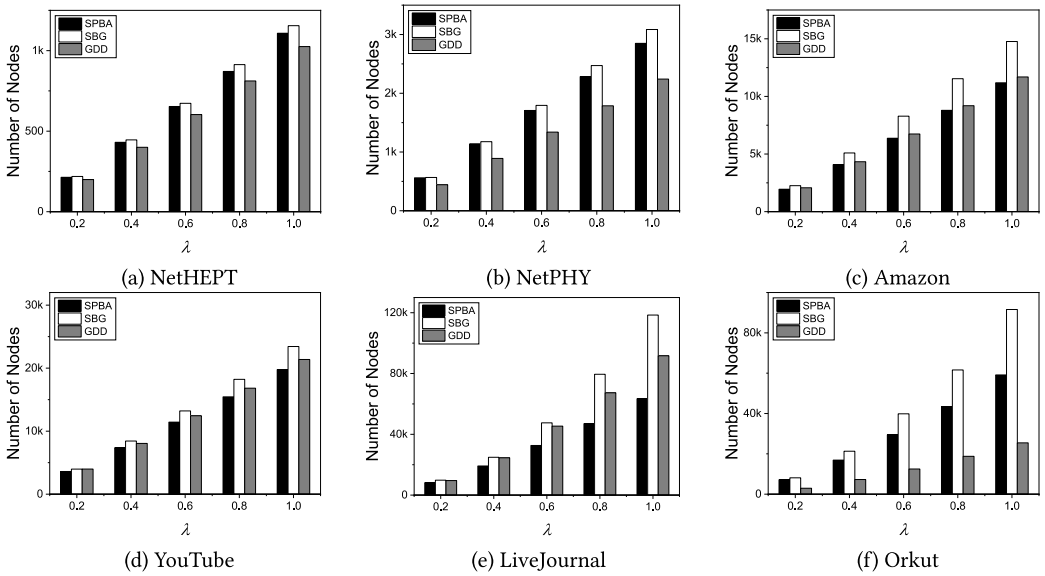
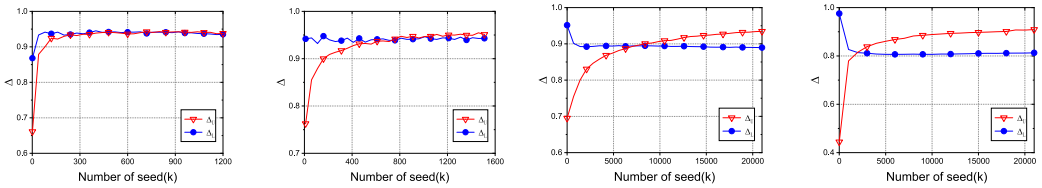


Fig. 12. The partial influence spread activated by groups with different acceptability λ.

YouTube are 3.41 and 19.2%, those in LiveJournal are 4.27 and 78.57%, and those in Orkut are 2.95 and 52.41%, respectively. It indicates that the larger the acceptability of the group opinion, the larger the spread of influence.

With the increase of the group consensus acceptability, the partial influence spread activated by groups increases significantly. To further study the influence diffusion process of the GCLT model, we compare the number of nodes activated by the groups with different λ. The results in Figure 12 show that the nodes activated by groups increase significantly with the increase of λ. Moreover,



(a) NetHEPT (b) NetPHY (c) Amazon (d) YouTube

Fig. 13. The approximation ratios of the upper and lower bounds to the objective function.

the increase of our *SBG* is larger than that of SPBA and GDD. For example, the number of nodes with $\lambda = 1.0$ is about 7.8 and 9.6 times greater than that with $\lambda = 0.2$ for the SPBA and the GDD in the LiveJournal dataset. And the improvement is about 12.1 times for our SBG. The results show that SBG can better integrate group consensus. Furthermore, the ratio of activated nodes increases with that of λ. This indicates that the increased nodes activated by groups can further spread more influence to other nodes and activate more groups.

In this article, we use λ as the acceptability of group consensus at the individual level, representing the conformity of an individual. The group effect can also be reflected in the network or group level, indicating the internal cohesion of the group. Therefore, the proposed GCLT model can be easily applied in different scenarios.

6.3 Approximation of Upper and Lower Bounds of the Objective Function

To verify the rationality of the upper and lower bounds of the objective function, we compare the upper and lower bounds with the original objective function. The evaluating indicators of the experiment are $\Delta_U = \frac{I(S_U)}{U(S_U)}$ and $\Delta_L = \frac{L(S_L)}{I(S_L)}$, which represent the ratios of the upper and lower bounds to the objective function. If the two values are close to 1, it indicates that the upper and lower bounds are rational. We sample a large number of URR sets and LRR sets to ensure that the errors are small. And we use the Monte Carlo simulation method to estimate the objective function $I(\cdot)$. We ensure that the errors are limited to 0.01 with a probability 0.99. Figure 13 shows the approximation ratios when different numbers of seed nodes are selected. We have the following main findings.

The maximum value of the approximation ratio remains above 0.9 in most cases, indicating the rationality of the upper and lower bounds. As shown in Figure 13, with the number of selected seed nodes increases, the value of Δ_L decreases while the value of Δ_U increases. This is because the LRR set selects the nodes that can influence the source node singly, while the URR set selects the nodes while assuming other nodes can block the negative nodes. When more seeds are selected, it's more like the case of URR. The maximum value of Δ_L and Δ_U remains above 0.9 in most cases, which shows the rationality of the upper and lower bounds. With the constraint of the upper and lower bounds of the objective function, the SBG algorithm can obtain better solutions by selecting seed sets of different sizes.

6.4 Experiment of the Group Consensus

Group consensus plays a very important role in the influence propagation process. To study the effect of group consensus, we analyze the influence propagation in the GCLT model and the CLT model with different sizes of negative seeds. In the experiment, we select different numbers of groups. In these selected groups, we randomly select $\log_2 k$ nodes as negative seeds, where k is the size of the group. The proportions of the groups that selected negative seeds are {0.2, 0.4, 0.6,

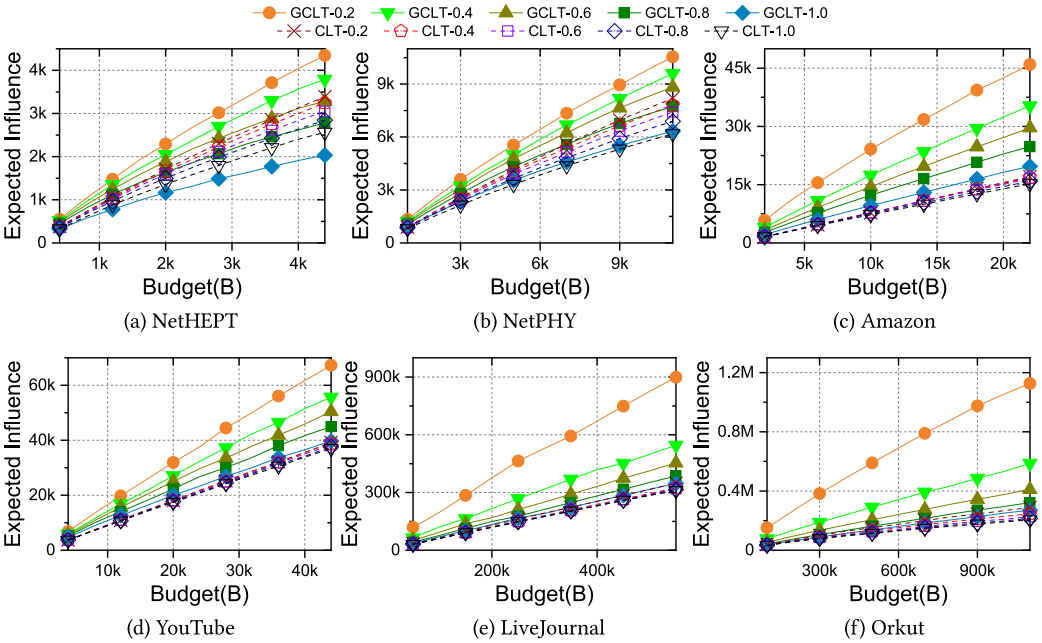


Fig. 14. Influence spread with different sizes of negative seeds in the GCLT model and the CLT model.

0.8, 1.0}. We use SBG to get positive seeds with different budgets and negative seeds, and contrast the influence spread in the GCLT model and the CLT model.

The increase of negative seeds decreases the influence spread in the GCLT model, indicating group consensus greatly impacts the propagation process. As shown in Figure 14, under the same budget, the positive influence spread in the GCLT model fluctuates more with the number of negative seeds, compared with the CLT model. It illustrates that the change in the number of negative seeds has a greater impact on the influence spread in the GCLT model. This is because the group consensus-reaching process in the GCLT model makes the nodes in the group more likely to be affected. When there are few negative nodes, the positive influence is easy to spread in the GCLT model. However, when there are more negative nodes, the positive influence is more easily suppressed by the negative influence. This experiment shows that group consensus greatly impacts the propagation of various information. Therefore, it is necessary to consider and study the group consensus in social networks with a group structure.

The average stopping step of the GCLT model is longer than that of the CLT model, indicating it can extract the potential connections between nodes. We calculate the average stop steps of influence diffusion under different budgets on the Amazon and NetPHY datasets. The results are shown in Table 3. It shows that the average stopping step of the GCLT model is about 1–3 steps longer than that of the CLT model, which indicates that the association path between different nodes in the GCLT model is longer. In the GCLT model, the influence can spread to further nodes, which reflects the ability of the GCLT model to mine the potential connections between nodes.

7 Conclusion

This article strives to integrate group consensus to maximize competitive influence in OSNs. Specifically, we propose a new influence diffusion model, the GCLT model, to study the problem of BCIM. We investigate the BCIM problem under the GCLT model, which is NP-hard, and the objective

Table 3. Average Stop Steps of Influence Propagation

Amazon	Average Stop Steps		NetPHY	Average Stop Steps	
Budget	GCLT	CLT	Budget	GCLT	CLT
50K	27.68	23.40	2K	23.13	21.71
100K	24.95	22.54	4K	23.06	21.91
150K	23.99	21.86	6K	22.43	21.22
200K	23.06	21.33	8K	22.15	21.27
250K	22.03	21.21	10K	21.80	20.73

function is neither submodular nor supermodular. To solve BCIM, we propose the SBG algorithm by applying the Sandwich Approximation framework, in which we provide an approximate solution to the lower bound and the upper bound by the proposed OPIM-B algorithm and select the seed set of solutions that achieves the best influence spread in Monte Carlo simulations. We also propose two strategies to optimize the time efficiency of SBG. The experiments show that the proposed algorithm is efficient and effective. Moreover, the GCLT model can explore more connections between nodes by integrating group consensus.

Our research still has some limitations that are worth further study. First, we mainly consider scenarios with non-overlapping groups, and more scenarios with overlapping groups can be further studied. Future research can be done to deal with overlapping groups. Second, our idea of integrating group consensus can also be applied to other diffusion models, as long as they can consider group effects. Third, studying the optimal lower and upper bounds of the objective function is also promising, especially the approximation of the upper and lower bounds of the objective function in different datasets. Last but not least, many factors may impact the group consensus reaching in real life, such as the personality of members, the type of relationships in the group, and the preference scenarios [47]. Those factors and their impact on information diffusion are also worth further study.

References

[1] Suman Banerjee, Mamata Jenamani, and Dilip Kumar Pratihar. 2019. ComBIM: A community-based solution approach for the budgeted influence maximization problem. *Expert Systems with Applications* 125 (2019), 1–13.

[2] Shishir Bharathi, David Kempe, and Mahyar Salek. 2007. Competitive influence maximization in social networks. In *International Workshop on Web and Internet Economics*. Springer, 306–311.

[3] Song Bian, Qintian Guo, Sibo Wang, and Jeffrey Xu Yu. 2020. Efficient algorithms for budgeted influence maximization on massive social networks. *Proceedings of the VLDB Endowment* 13, 9 (May 2020), 1498–1510. DOI: https://doi.org/10.14778/3397230.3397244

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

[5] Allan Borodin, Yuval Filmus, and Oren Joel. 2010. Threshold models for competitive influence in social networks. In *International Workshop on Internet and Network Economics*. Springer, 539–550.

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

[7] Ceren Budak, Divyakant Agrawal, and Amr El Abbadi. 2011. Limiting the spread of misinformation in social networks. In *International Conference on World Wide Web*, 665–674.

[8] Taotao Cai, Qi Lei, Quan Z. Sheng, Ningning Cui, Shuiqiao Yang, Jian Yang, Wei Emma Zhang, and Adnan Mahmood. 2024. Reconnecting the estranged relationships: Optimizing the influence propagation in evolving networks. *IEEE Transactions on Knowledge and Data Engineering* 36, 5 (2024), 2151–2165. DOI: https://doi.org/10.1109/TKDE.2023.3316268

[9] Tim Carnes, Chandrashekhar Nagarajan, Stefan M. Wild, and Anke Van Zuylen. 2007. Maximizing influence in a competitive social network: A follower's perspective. In *International Conference on Electronic Commerce*, 351–360.

[10] Wei Chen, Yajun Wang, and Siyu Yang. 2009. Efficient influence maximization in social networks. In *15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 199–208.

[11] David Contreras, Maria Salamó, and Ludovico Boratto. 2021. Integrating collaboration and leadership in conversational group recommender systems. *ACM Transactions on Information Systems* 39, 4, Article 41 (Aug. 2021), 32 pages. DOI: https://doi.org/10.1145/3462759

[12] Paul Dagum, Richard Karp, Michael Luby, and Sheldon Ross. 2000. An optimal algorithm for Monte Carlo estimation. *SIAM Journal on Computing* 29, 5 (2000), 1484–1496.

[13] Yajun Dai, Wenjun Jiang, and Kenli Li. 2018. Group-based competitive influence maximization. In *2018 IEEE Ubiquitous Intelligence & Computing*. IEEE, 999–1006.

[14] Morris H. DeGroot. 1974. Reaching a consensus. *Journal of the American Statistical Association* 69, 345 (1974), 118–121.

[15] Güney, E. 2019. On the optimal solution of budgeted influence maximization problem in social networks. *Operational Research* 19, 3 (2019), 817–831.

[16] Smita Ghosh, Tiantian Chen, and Weili Wu. 2025. Enhanced group influence maximization in social networks using deep reinforcement learning. *IEEE Transactions on Computational Social Systems* 12, 2 (2025), 573–585. DOI: https://doi.org/10.1109/TCSS.2024.3459853

[17] Teresa González-Arteaga, Rocío de Andrés Calle, and Francisco Chiclana. 2016. A new measure of consensus with reciprocal preference relations: The correlation consensus degree. *Knowledge-Based Systems* 107 (2016), 104–116.

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

[19] Xinran He, Guojie Song, Wei Chen, and Qingye Jiang. 2012. Influence blocking maximization in social networks under the competitive linear threshold model. In *2012 Siam International Conference on Data Mining*. SIAM, 463–474.

[20] Wenjing Hong, Chao Qian, and Ke Tang. 2021. Efficient minimum cost seed selection with theoretical guarantees for competitive influence maximization. *IEEE Transactions on Cybernetics* 51, 12 (2021), 6091–6104.

[21] Janusz Kacprzyk. 1986. Group decision making with a fuzzy linguistic majority. *Fuzzy Sets and Systems* 18, 2 (1986), 105–118.

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

[23] Samir Khuller, Anna Moss, and Joseph Seffi Naor. 1999. The budgeted maximum coverage problem. *Information Processing Letters* 70, 1 (1999), 39–45.

[24] Sunil Kumar Meena, Shashank Sheshar Singh, and Kuldeep Singh. 2024. Cuckoo search optimization-based influence maximization in dynamic social networks. *ACM Transactions on the Web* 18, 4, Article 49 (Oct. 2024), 25 pages. DOI: https://doi.org/10.1145/3690644

[25] Jure Leskovec, Andreas Krause, Carlos Guestrin, Christos Faloutsos, Jeanne VanBriesen, and Natalie Glance. 2007. Cost-effective outbreak detection in networks. In *13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 420–429.

[26] Yuchen Li, Ju Fan, Yanhao Wang, and Kian-Lee Tan. 2018. Influence maximization on social graphs: A survey. *IEEE Transactions on Knowledge and Data Engineering* 30, 10 (2018), 1852–1872. DOI: https://doi.org/10.1109/TKDE.2018.2807843

[27] Yishi Lin and John C. S. Lui. 2015. Analyzing competitive influence maximization problems with partial information: An approximation algorithmic framework. *Performance Evaluation* 91 (2015), 187–204. Special Issue: Performance 2015.

[28] Wei Lu, Wei Chen, and Laks V. S. Lakshmanan. 2015. From competition to complementarity: Comparative influence diffusion and maximization. *Proceedings of the VLDB Endowment* 9, 2 (Oct. 2015), 60–71. DOI: https://doi.org/10.14778/2850578.2850581

[29] Wei Lu and Laks V. S. Lakshmanan. 2012. Profit maximization over social networks. In *2012 IEEE 12th International Conference on Data Mining*. IEEE, 479–488.

[30] Zongqing Lu, Yonggang Wen, and Guohong Cao. 2014. Information diffusion in mobile social networks: The speed perspective. In *IEEE Infocom*, 1932–1940.

[31] Sunil Kumar Meena, Shashank Sheshar Singh, and Kuldeep Singh. 2024. DCDIMB: Dynamic community-based diversified influence maximization using bridge nodes. *ACM Transactions on the Web* 18, 4, Article 47 (Oct. 2024), 32 pages. DOI: https://doi.org/10.1145/3664618

[32] Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations (ICLR '13)*. Yoshua Bengio and Yann LeCun (Eds.). Retrieved from http://arxiv.org/abs/1301.3781

[33] Evgeny Morozov. 2009. Swine flu: Twitter's power to misinform. *National Public Radio* (2009). https://www.npr.org/2009/04/28/103562240/swine-flu-twitters-power-to-misinform

[34] Huy Nguyen and Rong Zheng. 2013. On budgeted influence maximization in social networks. *IEEE Journal on Selected Areas in Communications* 31, 6 (2013), 1084–1094.

[35] H. T. Nguyen, M. T. Thai, and T. N. Dinh. 2016. Stop-and-stare: Optimal sampling algorithms for viral marketing in billion-scale networks. In *The 2016 International Conference*.

[36] Nam P. Nguyen, Guanhua Yan, and My. T. Thai. 2013. Analysis of misinformation containment in online social networks. *Computer Networks* 57, 10 (2013), 2133–2146.

[37] Iván Palomares, Luis Martínez, and Francisco Herrera. 2014. A consensus model to detect and manage noncooperative behaviors in large-scale group decision making. *IEEE Transactions on Fuzzy Systems* 22, 3 (2014), 516–530. DOI: https://doi.org/10.1109/TFUZZ.2013.2262769

[38] Thomas F. Pettigrew, Linda R. Tropp, Ulrich Wagner, and Oliver Christ. 2011. Recent advances in intergroup contact theory. *International Journal of Intercultural Relations* 35, 3 (2011), 271–280.

[39] Canh V. Pham, Hieu V. Duong, Huan X. Hoang, and My. T. Thai. 2019. Competitive influence maximization within time and budget constraints in online social networks: An algorithmic approach. *Applied Sciences* 9, 11 (2019), 2274.

[40] Akrati Saxena, George Fletcher, and Mykola Pechenizkiy. 2024. FairSNA: Algorithmic fairness in social network analysis. *ACM Computing Surveys* 56, 8, Article 213 (Apr. 2024), 45 pages. DOI: https://doi.org/10.1145/3653711

[41] Guojie Song, Xiabing Zhou, Yu Wang, and Kunqing Xie. 2015. Influence maximization on large-scale mobile social network: A divide-and-conquer method. *IEEE Transactions on Parallel and Distributed Systems* 26, 5 (2015), 1379–1392.

[42] Jing Tang, Xueyan Tang, Xiaokui Xiao, and Junsong Yuan. 2018. Online processing algorithms for influence maximization. In *2018 International Conference on Management of Data (SIGMOD '18)*. ACM, New York, NY, 991–1005. DOI: https://doi.org/10.1145/3183713.3183749

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

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

[45] Konstantinos Theocharidis, Panagiotis Karras, Manolis Terrovitis, Spiros Skiadopoulos, and Hady W. Lauw. 2024. Adaptive content-aware influence maximization via online learning to rank. *ACM Transactions on Knowledge Discovery from Data* 18, 6, Article 146 (Apr. 2024), 35 pages. DOI: https://doi.org/10.1145/3651987

[46] V. A. Traag, Dooren P. Van, and Y. Nesterov. 2011. Narrow scope for resolution-limit-free community detection. *Physical Review. E, Statistical, Nonlinear, and Soft Matter Physics* 84, 2 (2011), 016114.

[47] Thi Ngoc Trang Tran, Alexander Felfernig, and Viet-Man Le. 2024. An overview of consensus models for group decision-making and group recommender systems. *User Modeling and User-Adapted Interaction* 34 (2024), 489–547. DOI: https://doi.org/10.1007/s11257-023-09380-z

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

[49] Yanhao Wang, Yuchen Li, Ju Fan, and Kian-Lee Tan. 2018. Location-aware influence maximization over dynamic social streams. *ACM Transactions on Information Systems* 36, 4, Article 43 (Jul. 2018), 35 pages. DOI: https://doi.org/10.1145/3230871

[50] Jiadong Xie, Zehua Chen, Deming Chu, Fan Zhang, Xuemin Lin, and Zhihong Tian. 2024. Influence maximization via vertex countering. *Proceedings of the VLDB Endowment* 17, 6 (May 2024), 1297–1309. DOI: https://doi.org/10.14778/3648160.3648171

[51] Qinghan Xue, Jiaqi Song, and Xingqin Qi. 2024. DASH: A novel method for dynamically selecting key nodes to spread information rapidly under the graph burning model. *Physics Letters A* 528 (2024), 130058. DOI: https://doi.org/10.1016/j.physleta.2024.130058

[52] Qian Yan, Hao Huang, Yunjun Gao, Wei Lu, and Qinming He. 2017. Group-level influence maximization with budget constraint. In *Database Systems for Advanced Applications*. Springer International Publishing, Cham, 625–641.

[53] Jaewon Yang and Jure Leskovec. 2012. Defining and evaluating network communities based on Ground-Truth. In *2012 IEEE 12th International Conference on Data Mining*, 745–754. DOI: https://doi.org/10.1109/ICDM.2012.138

[54] Shiqi Zhang, Yiqian Huang, Jiachen Sun, Wenqing Lin, Xiaokui Xiao, and Bo Tang. 2023. Capacity constrained influence maximization in social networks. In *29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM, 3376–3385. DOI: https://doi.org/10.1145/3580305.3599267

[55] Shiqi Zhang, Jiachen Sun, Wenqing Lin, Xiaokui Xiao, and Bo Tang. 2022. Measuring friendship closeness: A perspective of social identity theory. In *31st ACM International Conference on Information & Knowledge Management (CIKM '22)*. ACM, New York, NY, 3664–3673. DOI: https://doi.org/10.1145/3511808.3557076

[56] Wentao Zhang, Xinyi Gao, Ling Yang, Meng Cao, Ping Huang, Jiulong Shan, Hongzhi Yin, and Bin Cui. 2024. BIM: Improving graph neural networks with balanced influence maximization. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, 2931–2944. DOI: https://doi.org/10.1109/ICDE60146.2024.00228

[57] Yuting Zhong and Longkun Guo. 2020. Group influence maximization in social networks. In *International Conference on Computational Data and Social Networks*. Springer, 152–163.

[58] Jianming Zhu, Smita Ghosh, and Weili Wu. 2019. Group influence maximization problem in social networks. *IEEE Transactions on Computational Social Systems* 6, 6 (2019), 1156–1164. DOI: https://doi.org/10.1109/TCSS.2019.2938575

[59] Yuqing Zhu, Deying Li, and Zhao Zhang. 2016. Minimum cost seed set for competitive social influence. In *the IEEE International Conference on Computer Communications (INFOCOM '16)*, IEEE, 1–9. DOI: https://doi.org/10.1109/INFOCOM.2016.7524472

Appendix

A Proof of Theorem 4.1

We use the inductive method to illustrate the equivalent of two models in every step t. Let S_P^t and S_N^t be the *P_active* and the *N_active* node sets of the GCLT model in step t, respectively. Similarly, $S_P^{'t}$ and $S_N^{'t}$ are node sets of the GCLE model in step t. In step $t = 0$, we have $S_P^t = S_P^{'t}$ and $S_N^t = S_N^{'t}$.

For the GCLT model, we first consider the groups that have not been activated at step t. Based on the activation rules of groups in the GCLT model, a group C will be positively activated at step $t + 1$ with probability:

$$\begin{aligned} P_1^{t+1}(C) &= \left(1 - \sum_{u \in S_N^t} w^-(u, C)\right) \sum_{u \in S_P^t} w^+(u, C) + \frac{1}{2} \sum_{u \in S_N^t} w^-(u, C) \sum_{u \in S_P^t} w^+(u, C) \\ &= \left(1 - \frac{1}{2} \sum_{u \in S_N^t} w^-(u, C)\right) \sum_{u \in S_P^t} w^+(u, C). \end{aligned} \tag{A1}$$

After the group C is activated, the nodes in C will be activated with probability λ:

$$P_2^{t+1}(v) = \lambda P_1^{t+1}(C). \tag{A2}$$

Then, the remaining *inactive* nodes will be positively activated at step $t + 1$ with probability:

$$P_3^{t+1}(v) = \left(1 - \frac{1}{2} \sum_{u \in S_N^t} w^-(u, v)\right) \sum_{u \in S_P^t} w^+(u, v). \tag{A3}$$

In the GCLE model, an *inactive* group node x will be positively activated at step $t + 1$ with probability:

$$\begin{aligned} P_1^{'t+1}(x) &= \left(1 - \sum_{u \in S_N^{'t}} w^-(u, x)\right) \sum_{u \in S_P^{'t}} w^+(u, x) + \frac{1}{2} \sum_{u \in S_N^{'t}} w^-(u, x) \sum_{u \in S_P^{'t}} w^+(u, x) \\ &= \left(1 - \frac{1}{2} \sum_{u \in S_N^{'t}} w^-(u, x)\right) \sum_{u \in S_P^{'t}} w^+(u, x). \end{aligned} \tag{A4}$$

After the group node x is activated, x will activate the nodes that it connects to. The node v in the same group with x will be positively activated by x with probability:

$$P_2^{'t+1}(v) = \lambda P_1^{'t+1}(x). \tag{A5}$$

Then, the remaining *inactive* nodes will be positively activated at step $t + 1$ with probability:

$$P_3^{'t+1}(v) = \left(1 - \frac{1}{2} \sum_{u \in S_N^{'t}} w^-(u, v)\right) \sum_{u \in S_P^{'t}} w^+(u, v). \tag{A6}$$

When $S_N^t = S_N^{'t}$ and $S_P^t = S_P^{'t}$, there are $P_1^{t+1}(C) = P_1^{'t+1}(x)$, $P_2^{t+1}(v) = P_2^{'t+1}(v)$, and $P_3^{t+1}(v) = P_3^{'t+1}(v)$. The negatively activate probabilities are similar. Because $S_P^0 = S_P^{'0}$ and $S_N^0 = S_N^{'0}$, by step-by-step induction, the distribution of *P_active* and *N_active* sets in the GCLE model are the same as the GCLT model at each step.

Received 14 February 2025; revised 15 July 2025; accepted 11 September 2025