Composite Community-Aware Diversified Influence Maximization With Efficient Approximation

Jianxiong Guo†, Member, IEEE, ACM, Qiufen Ni†, Weili Wu†, Senior Member, IEEE, and Ding-Zhu Du

Abstract—Influence Maximization (IM) is a well-known topic in mobile networks and social computing that aims to find a small subset of users that maximize the influence spread through an online information cascade. Recently, some cautious researchers have paid attention to the diversity of information dissemination, especially community-aware diversity, and formulated the diversified IM problem. Diversity is ubiquitous in many real-world applications, but these applications are all based on a given community structure. In social networks, we can form heterogeneous community structures for the same group of users according to different metrics. Therefore, how to quantify diversity based on multiple community structures is an interesting question. In this paper, we propose a Composite Community-Aware Diversified IM (CC-DIM) problem, which aims to select a seed set to maximize the influence spread and the composite diversity over all possible community structures under consideration. To address the NP-hardness of the CC-DIM problem, we adopt the technique of reverse influence sampling and design a random Generalized Reverse Reachable (G-RR) set to estimate the objective function. The composition of a random G-RR set is much more complex than the RR set used for the IM problem, which will lead to the inefficiency of traditional sampling-based approximation algorithms. Because of this, we further propose a two-stage algorithm, Generalized HIST (G-HIST). It can not only return a \((1 - 1/e - \epsilon)\) approximate solution with at least \((1 - \delta)\) probability but also improve the efficiency of sampling and ease the difficulty of searching by significantly reducing the average size of G-RR sets. Finally, we evaluate our proposed G-HIST on real datasets against existing algorithms. The experimental results show the effectiveness of our proposed algorithm and its superiority over other baseline algorithms.

Index Terms—Influence maximization, social networks, composite diversity, reverse sampling, approximation algorithm.

I. INTRODUCTION

ONLINE social networks (OSNs) are connected by hundreds of millions of mobile devices through social media, which has become a popular platform for people to express their views, for companies to promote their products, and for governments to spread their policies. With the rapid development of the mobile internet, there are more than 1.52 billion daily users on Facebook and 321 million monthly users active on Twitter, prompting the study of the Influence Maximization (IM) problem. It selects a small subset of influential users (seed nodes) in social networks, convinces them to adopt one thing (product, service, or opinion), and utilizes the “word-of-mouth” effect to activate other users in social networks through an online information cascade. Kempe et al. [1] formally defined the IM problem as a combinatorial optimization problem, which aims to select a size-\(k\) seed set such that the expected number of activated nodes can be maximized in the social network. Subsequently, a series of variant optimization problems based on the IM problem for different real-world applications were developed, such as topic-aware IM [2], [3], time-aware IM [4], [5], location-aware IM [6], [7], and target-aware IM [8], [9].

Both the IM problem and its variant problems are only concerned with how to maximize the total number of activated nodes across the network. They do not care who was the activated node, and thus, the diversity of activated nodes did not attract enough attention in the current research. For example, to consider community-aware diversity, users on social media from different communities usually represent different kinds of people, and their classification metrics include age, gender, occupation, and income. For an organization to advertise its ideas or products, it usually strives for diverse followers from different communities, so as to increase its influence more effectively. In addition, the diversity of recommendations is also an important criterion to measure the quality of recommendation systems [10], [11]. Thus, diversity could be an advantage, as it is not a good idea to focus all resources in one area. To the best of our knowledge, only five studies [12], [13] [14], [15], [16] have considered diversity in the IM problem. They all took community-aware diversity as a specific example and, based on the initial work [12], tried to maximize the expected value of the influence spread and influence diversity.
In the Community-Aware Diversified IM problem, it is necessary to partition a given social graph into communities in advance to achieve diversity in this community structure. A community structure is usually formulated according to a certain metric, which can effectively measure the distance between two nodes. If the metric used to partition a social graph is users’ occupations, then the diversity here will be occupation-oriented. However, in real-world applications, there is often more than one metric. Sometimes, we want to consider diversity based on several metrics at the same time. Let us first look at the following example.

Example 1: For the government to spread its policies, it can select some influential celebrities to publicize and spread influence across the network. At this time, the government should not only allow as many listeners as possible to receive the message but also consider diversity, including age, occupation, income, social class, etc. In this case, each factor represents a kind of community structure. As shown in Fig. 1, the government can spread its policies in various communities in a balanced way and can freely adjust the weight of different kinds of communities so that it can pay more attention to a certain type of group.

As shown in Example 1, only one kind of community structure is obviously insufficient, and the previous diversified IM problem cannot cover this scenario based on multiple metrics.

Therefore, we propose a Composite Community-Aware Diversified Influence Maximization (CC-DIM) problem in this paper, which perfectly matches multiple metrics of the community partition. For the CC-DIM problem, we first formulate different community structures according to the metrics we consider and then select a size-$k$ seed set such that the weighted sum of the expected number of activated nodes and the expected diversity of activated nodes, where the expected diversity is the average over all possible community structures based on different metrics. Then, we prove that the CC-DIM problem is NP-hard but that its objective function is monotone and submodular. Unfortunately, similar to computing the influence spread in the IM problem [17], [18], it is #P-hard as well to compute our objective function, which is even more difficult. If using the greedy hill-climbing algorithm with Monte Carlo simulations to estimate the objective function, then the computational cost will be unacceptably high. To improve its computational efficiency, many methods based on the technique of reverse influence sampling (RIS) [19] have been utilized to solve the IM problem. Here, we design a novel sampling method, called the random Generalized Reverse Reachable (G-RR) set, to unbiasedly estimate the objective function of the CC-DIM problem. However, the processes of sampling and searching based on random G-RR sets are much more complex and time-consuming than those based on random RR sets for the IM problem because of the diversity in multiple community structures. Thus, based on random G-RR sets and probabilistic analysis, we further propose a two-stage algorithm, called Generalized HIST (G-HIST), which includes sentinel set selection and remaining set selection. G-HIST selects a small-size sentinel set by a small number of random G-RR sets in the first stage and then utilizes the sentinel set to significantly reduce the average size of random G-RR sets in the second stage. Through detailed theoretical analysis and experimental verification, we prove that the memory consumption and running time are greatly improved by G-HIST because of its compressed sampling, where the approximation guarantee will not be affected. The contributions of this paper are summarized as follows.

- To the best of our knowledge, we are the first to consider the diversity of multiple community structures according to different metrics, propose the CC-DIM problem, and prove its hardness, monotonicity, and submodularity.
- To tackle intractability, we design a random G-RR set and an unbiased estimator of the objective function under the special case where the utility is linear. Then, we propose a G-HIST algorithm to further reduce the memory consumption and running time, which can return a $(1 - 1/e - \varepsilon)$ approximate solution with at least $(1 - \delta)$ probability.
- We conduct intensive simulations based on real-world social datasets. By comparing our G-HIST with state-of-the-art baselines, the experimental results validate the effectiveness of our proposed sampling and algorithm in approximate performance and efficiency.
- We extend our sampling-based estimation and algorithm to solve the CC-DIM problem when the utility is an arbitrary concave function, and then we point out feasible solutions and research directions in the future.

Organization: In Sec. II, we summarize the works related to this paper. We then introduce our CC-DIM problem and its basic properties in Sec. III and the sampling techniques used for estimation in Sec. IV. In Sec. V, we elaborate on the G-HIST algorithm and its corresponding theoretical analysis. Experiments and discussions are presented in Sec. VI and Sec. VII. Finally, Sec. VIII concludes this paper.

II. RELATED WORKS

A. Influence Maximization

Kempe et al. [1] first formulated the IM problem and defined it as a combinatorial optimization problem. They proposed
two classic diffusion models, the Independent Cascade (IC) model and the Linear Threshold (LT) model, and proved that the IM problem is NP-hard and that the influence spread is monotone and submodular. Given a seed set, it is #P-hard to compute the influence spread under the IC model [17] and LT-model [18]. Thus, the greedy hill-climbing algorithm can return a $(1-1/e-\epsilon)$ approximate solution with Monte Carlo simulations. Borgs et al. [19] first proposed the RIS technique to reduce the running time, but they did not give a feasible algorithm and a strict theoretical argument. Subsequently, a plethora of research works focused on following the RIS to further improve efficiency, such as TIM/TIM+ [20], IMM [21], SSA/D-SSA [22], and OPIM-C [23]. Along this line, it could run in $O(k(n+m)\log n/\epsilon^2)$ expected time and return a $(1-1/e-\epsilon)$ approximate solution with at least $1-1/n$ probability. Recently, Guo et al. [24], [25] proposed a Hit-and-Stop (HIST) algorithm to tackle the scalability issue in high-influence networks by reducing the average size of random RR sets without losing the approximation guarantee. For our CC-DIM problem, we learn from the idea of HIST and formulate our G-HIST algorithm because the size of random G-RR sets will be much larger than that of random RR sets from existing multiple community structures. However, our G-HIST is not a trivial revision of HIST, and we extend it to a variant optimization problem in social networks.

B. Diversified Influence Maximization

Based on the above-mentioned community structures in social networks, the diversity of influence spread has become the inherent demand of viral marketing, which is a typical application of the IM problem. Tang et al. [12] first defined the diversified IM problem as a combinatorial optimization problem that aims to maximize the weighted sum of the influence spread and diversity while designing an algorithm to solve it. Zhang [13] adopted three commonly used utilities in economics to quantify the diversity of communities. Li et al. [14] proposed a metric to measure the community-based diversified influence and designed two tree-based heuristic algorithms to reduce the computational cost. Wang et al. [15] utilized the IMM algorithm to effectively address the diversified IM problem by sampling with a theoretical guarantee. Zhang et al. [16] exploited a Graph Neural Network (GNN)-based method, GRAIN, to combine the diversified IM and greedy algorithm into a unified framework, which significantly improved the efficiency of data selection.

C. Discussion

Even though the IM problem has been intensively studied and the diversified IM problem has been proposed recently, they still do not cover all the issues. First, there is no existing work that considers the multiplicity of diversity, namely, the diversity of multiple kinds of community structures, but this is meaningful in real applications. Second, how to design an effective approximation algorithm for the complex modeling problem in social networks has not been well solved. This reflects the necessity of our further research in this direction, and solving these two challenges is the main contribution of this paper.

| Notation | Description |
|----------|-------------|
| $G = (V, E)$ | An instance of the social network |
| $g$ | A realization sampled from a given model |
| $S \subseteq V$ | The seed set |
| $\sigma(S)$ | Expected influence spread of $S$ |
| $I_g(S)$ | The node set reachable from $S$ in $g$ |
| $Q$ | Metric set; $q \in Q$ is a metric |
| $c_q(G)/c_q$ | Community Partition based on metric $q$ |
| $\psi(S; c_q)$ | Diversified function of $S$ under $c_q$ |
| $\phi(S)$ | Composite diversified function of $S$ |
| $\gamma_q(x)$ | Utility of $S$ for each community $C^q \in C_q$ |
| $h_{q, \beta}(x)$ | A monotone and concave function w.r.t. $x$ |
| $f(S)$ | Objective function of CC-DIM problem |
| $\mathcal{R}$ | Each $\mathcal{R}_i \in \mathcal{R}$ is random G-RR set |
| $\hat{f}(S; \mathcal{R})$ | An estimation of $f(S)$ on $\mathcal{R}$ |
| $\epsilon/\delta/\eta_i/\eta_n$ | Parameters used to bound approx. error |

III. PROBLEM FORMULATION

In this section, we define our CC-DIM problem from the basic definitions of diffusion models, community structures, and the IM problem. The frequently used notations are given in Table I.

A. Preliminaries

Let $G = (V, E)$ be a directed graph with a node set $V = \{v_1, v_2, \cdots, v_n\}$ and an edge set $E = \{e_1, e_2, \cdots, e_m\}$. In social networks, each node $v \in V$ represents a user and each edge $(u, v) \in E$ represents the relationship, e.g., friendship, between $u$ and $v$. For each edge $(u, v) \in E$, we say that $u$ is the in-neighbor of $v$ and $v$ is the out-neighbor of $u$. For each node $v \in V$, we denote by $N^-(v)$ the set of its in-neighbors and $N^+(v)$ the set of its out-neighbors.

In information diffusion, we consider a user to be active if she accepts (is activated by) the information cascade from her in-neighbors or if she is selected as a seed. The information cascade can be given by a predefined diffusion model, such as the Independent Cascade (IC) model [1]. Given a seed set $S \subseteq V$, the IC model is a discrete-time stochastic cascade process shown as follows: (1) At timestamp 0, all nodes in $S$ are activated and other nodes in $V\setminus S$ are inactive, where a node keeps active once it is activated; (2) If a node $u$ is activated at timestamp $t$, it has one chance to activate its inactive out-neighbor $v$ with the probability $p_{uv}$ at timestamp $t + 1$, after which it cannot activate any nodes; and (3) The information diffusion terminates when no more inactive nodes can be activated in the subsequent timestamp.

B. Influence Maximization (IM)

The traditional IM problem involves finding a seed set $S \subseteq V$ such that its influence spread $\sigma(S)$ can be maximized, which is the expected number of active nodes after the diffusion terminates. To mathematically define the influence spread,
we first introduce a concept called “realization”. A realization
\( g = (V, E_g) \) with \( E_g \subseteq E \), is a subgraph sampled according to the diffusion model. For example, in the IC model, each edge \((u, v) \in E\) will be independently contained in \( E_g \) with the probability \( p_{uv} \). An edge in \( E_g \) is called a “live edge” in realization \( g \). Thus, the probability of realization \( g \) sampled from \( G \) under the IC model is \( \Pr[g] = \prod_{e \in E} p_{e} \prod_{e \notin E} (1-p_{e}) \). There are a total of \( 2^{m} \) possible realizations. The influence cascade on a realization becomes deterministic instead of a stochastic process. As a result, the influence spread across the network can be regarded as the expected spread on all possible realizations. Now, the IM problem can be written in an expectation form and formally defined as follows.

**Definition 1 (Influence Maximization):** Given a social graph \( G = (V, E) \), a diffusion model (IC model in this paper), and a budget \( k \), the IM problem aims to find a seed set \( S^* \), with at most \( k \) nodes, that can maximize the expected influence spread across the graph, i.e.

\[
S^* \in \arg \max_{|S| \leq k} \sigma(S) = \mathbb{E}_{g \sim G} [I_g(S)] = \sum_{g \in G} \Pr[g] \cdot |I_g(S)|, \tag{1}
\]

where \( G \) is the collection of all possible realizations sampled from a given diffusion model and \( I_g(S) \) is the node set that contains all nodes that can be reached from a node in \( S \) by the live edges in the realization \( g \). Here, the influence function \( \sigma \) is a set function. Given a set function \( f : 2^V \rightarrow \mathbb{R}_+ \) and any two subsets \( S \) and \( T \) with \( S \subseteq T \subseteq V \), we say it is monotone if \( f(S) \leq f(T) \) and submodular if \( f(S \cup \{v\}) - f(S) \geq f(T \cup \{v\}) - f(T) \) for any \( v \in V \setminus T \). The IM problem is NP-hard, and the influence function is monotone and submodular under the IC model. Under the size constraint, the greedy hill-climbing algorithm of iteratively choosing the node with maximum marginal gain approximates the optimal solution within a factor of \((1 - 1/e)\) [26]. However, given a seed set \( S \), it is \#P-hard to compute \( \sigma(S) \) under the IC model [17]. Thus, the hill-climbing algorithm can only return a \((1 - 1/e - \varepsilon)\) approximation within the \( \Omega(kmn \cdot poly(1/\varepsilon)) \) running time through Monte Carlo simulations [1].

### C. Composite Community-Aware Diversified IM

There are typically many communities in any given graph, and the community structure is an essential characteristic of social networks. In this way, the users can be divided into different groups according to a certain metric, and their communication within the group is dense but sparse between groups. Given a social network \( G = (V, E) \), we assume that it has a disjoint community structure \( \mathcal{C}(G) \) associated with \( G \), where \( \mathcal{C}(G) = \{C_1, C_2, \ldots, C_r\} \) is a partition of \( V \). That is \( V = \bigcup_{i=1}^{r} C_i \) and for any \( i, j \in \{1, 2, \ldots, r\} \), we have \( C_i \cap C_j = \emptyset \). However, when considering the diversified IM problem, the community structure can be partitioned based on different metrics, such as gender, age, race, interest, poor-rich disparity, and consuming behavior, for a variety of real applications. Thus, for the same optimization goal, community structure can be determined by different metrics. We denote by \( Q \) the metric set under our consideration. Each element \( q \in Q \) is a specific metric that can be used to partition the graph. In each metric \( q \in Q \), we define the community structure based on the metric \( q \) as \( \mathcal{C}_q = \mathcal{C}_q(G) = \{C_{q_1}^q, C_{q_2}^q, \ldots, C_{q_k}^q\} \), where \( r_q \) indicates that the graph can be divided into \( r_q \) communities under the metric \( q \). Now, we can formally define the composite diversified function as follows.

**Definition 2 (Composite Diversified Function):** Given a graph \( G = (V, E) \) and a metric set \( Q \), the composite diversified function of seed set \( S \) is defined as:

\[
\phi(S) = \sum_{q \in Q} w_q \cdot \psi(S; C_q) = \sum_{q \in Q} w_q \cdot \sum_{C_j^q \in \mathcal{C}_q} \bar{\psi}(S; C_j^q), \tag{3}
\]

where \( w_q \) is the weight of metric \( q \) that can ensure \( \sum_{q \in Q} w_q = 1 \). \( \psi(S; C_q) \) is a function that can quantify the diversity under the community structure \( C_q(G) \) generated by the metric \( q \), and \( \bar{\psi}(S; C_j^q) \) is the utility for each community \( C_j^q \) in \( C_q \).

Here, the weight \( w_q \) is the importance of the diversity of metric \( q \), and thus, the composite diversified function is a weighted average of the influence spread over all kinds of community structures based on multiple different metrics. To quantify the composite diversified function \( \phi(S) \), the question is transformed to how to define the utility \( \bar{\psi}(S; C_j^q) \) for each community \( C_j^q \) in \( C_q \). Intuitively, the utility \( \bar{\psi}(S; C_j^q) \) should reflect the degree of influence of \( S \) in the community \( C_j^q \). According to the above principles, we can define

\[
\bar{\psi}(S; C_j^q) = h_{q,j} \left( \mathbb{E}_{g \sim G} [I_{g}(S) \cap C_j^q] / |C_j^q| \right) \tag{4}
\]

where \( h_{q,j} : \mathbb{R}_+ \rightarrow \mathbb{R}_+ \) is a monotone and concave function with \( h_{q,j}(0) = 0 \). The \( h_{q,j}(x) \) is monotone, which implies that the utility increases as the influence in this community increases. It should be concave because we want to satisfy the diversity, that is, the contribution of increasing the influence of a certain community alone to the diversity will be weakened. Then, we have enough to define the Composite Community-Aware Diversified Influence Maximization (CC-DIM) problem as follows.

**Definition 3 (CC-DIM):** Given a graph \( G = (V, E) \), a metric set \( Q \), and a budget \( k \), the CC-DIM problem aims to select a subset \( S \) with \( |S| \leq k \) that can maximize the following objective function:

\[
f(S) = (1 - \lambda) \frac{\sigma(S)}{\sigma(V)} + \lambda \frac{\phi(S)}{\phi(V)}, \tag{6}
\]

where \( \lambda \in [0, 1] \) is an adjustable parameter that can balance the influence spread and the community-aware diversity.

**Example 2:** Here, the question is how to determine the value of \( \lambda \). For a commercial company, it wants to promote products and maximize profits. At this time, it needs to maximize the influence spread and does not care about the distribution of influence. Thus, \( \lambda \) should be zero. However, the government hopes that the promulgated policies can spread to people with different labels, such as incomes, genders, and interests. Thus, we first get three community structures partitioned by these three labels and then make the influence as evenly distributed as possible in these three community structures.
we extend our proposed method and algorithm to general errors through strict probability analysis. Finally in Sec. VII, we will present a constant approximation ratio by bounding the estimation algorithm, as shown in Sec. V, in polynomial time satisfying which transforms the original problem into a maximum expected utility objective function when the utility is linear in Sec. IV, mentioned in sampling techniques to accurately estimate our method is described in Fig. 2. We adopt the previously presented approximation ratio [26]. However, it is #P-hard to compute. In this section, we first introduce reverse influence sampling (RIS) [19] for the IM problem and then begin to discuss our generalized RIS method and propose the concept of G-RR sets for our CC-DIM problem.

A. Summary of the RIS Approach

The RIS approach is based on the concept of random reverse reachable (RR) sets. The IM problem, a random RR set \( R \) can be generated in three steps: (1) Uniformly select a node \( v \) from \( V \); (2) Randomly sample a realization \( g \) from \( \mathcal{G} \) under a given diffusion model; and (3) Collect the nodes that can reach \( v \) through a live path in realization \( g \). For each node \( u \in V \), the probability that it is contained in \( R \) generated by \( v \) equals the probability that \( v \) can activate \( u \). Here, we denote by \( R(v, g) \) a RR set generated from node \( v \) under realization \( g \), and then we have the following lemma.

Lemma 1 ([19]): Let \( S \subseteq V \) be a seed set and \( R \) be a random RR set under a given diffusion model, then we have
\[
\sigma(S) = n \cdot E_R[|S \cap R|] = n \cdot E_{v \sim V, g \sim \mathcal{G}}[|S \cap R(v, g)|],
\]
where \( |S \cap R| = 1 \) if \( S \cap R \neq \emptyset \), else \( |S \cap R| = 0 \).

B. Generalized Reverse Influence Sampling

For convenience, we assume a special case that the diversified utility is linear, and thus, we have \( h_{q,j}(x) = a_{q,j} \cdot x \) where \( a_{q,j} > 0 \) is an adjustable coefficient. When we find the proportion of \( E_{g \sim \mathcal{G}}[|I_g(S) \cap C_j^q|/|C_j^q|] \) is low, we can enlarge the coefficient \( a_{q,j} \) for diversity promotion. This is an important trick to help us achieve composite diversity. Although we use a simple linear function as an example here, it does not mean that our sampling-based method is only applicable to this special case. In the Sec. VII, we will elaborate on how to extend our method to more general cases, where \( h_{q,j}(x) \) is a family of more general concave functions. This shows that our method has generalization ability through a simple transformation.

However, the RIS cannot be directly applied to address our objective function because we need to consider the diversity under different metrics. To estimate \( \phi(S) \), we need to design a sampling method to compute the value of \( \tilde{\psi}(S; C_j^q) \) for each \( C_j^q \in C_q \) and \( q \in Q \) as a subroutine. Here, we have the following based on Eqn. (4),
\[
\tilde{\psi}(S; C_j^q) = a_{q,j} \cdot E_{g \sim \mathcal{G}}[|I_g(S) \cap C_j^q|/|C_j^q|].
\]
Then, the composite diversified function defined in Eqn. (3) can be rewritten as
\[
\phi(S) = \sum_{q \in Q} w_q \cdot \sum_{c_j^q \in C_q} a_{q,j} \cdot \sum_{g \in \mathcal{G}} \Pr[g] \cdot \frac{|I_g(S) \cap C_j^q|}{|C_j^q|} = \sum_{g \in \mathcal{G}} \Pr[g] \cdot \sum_{q \in Q} w_q \cdot \sum_{c_j^q \in C_q} a_{q,j} \cdot \frac{|I_g(S) \cap C_j^q|}{|C_j^q|}.
\]

D. Overview of Proposed Methods

Based on Theorem 2, the composite diversified function \( \phi(S) \) is monotone and submodular, and thus, the greedy hill-climbing algorithm can return a solution with \((1 - 1/e)\) approximation ratio [26]. However, it is #P-hard to compute this objective function. The framework of our proposed method is described in Fig. 2. We adopt the previously mentioned sampling techniques to accurately estimate our objective function when the utility is linear in Sec. IV, which transforms the original problem into a maximum coverage problem. Then, we design an effective approximation algorithm, as shown in Sec. V, in polynomial time satisfying a constant approximation ratio by bounding the estimation error through strict probability analysis. Finally in Sec. VII, we extend our proposed method and algorithm to general cases of the CC-DIM, where the utility is an arbitrary concave function.
Motivated by Eqn. (9), all communities $C^q_j$ for $C^q_j \in C_q$ and $q \in Q$ can share the same realization $g$ in computing the expectation. Therefore, we can formally define the concept of the random Generalized Reverse Reachable (G-RR) set. In our CC-DIM problem, a random G-RR set $\tilde{R}$ can be generated in the following three steps: (1) Uniformly select a node $v_{q,j}$ from $C^q_j$ for each community $C^q_j \in C_q$ and $q \in Q$, totally $\sum_{q \in Q} r_q$ nodes; (2) Randomly sample a realization $g$ from $G$ under a given diffusion model; and (3) For each node $v_{q,j}$ sampled from $C^q_j$, collect the nodes that can reach it through a live path in $g$. This is a random RR set $R(v_{q,j}, g)$, and thus, we have totally $\sum_{q \in Q} r_q$ random RR sets under the same realization. From above, a random G-RR set $\tilde{R}$ is a collection of random RR sets but they are under the same realization, which can be defined as

$$\tilde{R} = \{R(v_{q,j}, g) : C^q_j \in C_q \text{ and } q \in Q\}. \quad (10)$$

Then, we can build the relationship between the composite diversified function and the random G-RR set.

**Theorem 3:** Let $S \subseteq V$ be a seed set and $\tilde{R}$ be a random G-RR set under a given diffusion model. The function $\sigma(S)$ and $\phi(S)$ can be estimated as follows:

$$\sigma(S) = E_{\tilde{R}} \left[ \sum_{q \in Q} \sum_{C^q_j \in C_q} \frac{1}{|Q|} |C^q_j| \cdot \mathbb{I}(S \cap R(v_{q,j}, g)) \right], \quad (11)$$

$$\phi(S) = E_{\tilde{R}} \left[ \sum_{q \in Q} w_q \sum_{C^q_j \in C_q} a_{q,j} \cdot \mathbb{I}(S \cap R(v_{q,j}, g)) \right]. \quad (12)$$

**Proof:** For a seed set $S$, we have $E_{g \sim \tilde{g}}[I_g(S) \cap C^q_j] = |C^q_j| \cdot E_{g \sim \tilde{g}} \mathbb{I}(S \cap R(v, g))$ according to the Lemma 1. This indicates that

$$\sigma(S) = \sum_{C^q_j \in C_q} |C^q_j| \cdot E_{g \sim \tilde{g}}\mathbb{I}(S \cap R(v, g)) = \sum_{q \in Q} \sum_{C^q_j \in C_q} \frac{1}{|Q|} |C^q_j| \cdot E_{g \sim \tilde{g}}\mathbb{I}(S \cap R(v, g)) = E_{g \sim \tilde{g}} \left[ \sum_{q \in Q} \sum_{C^q_j \in C_q} \frac{1}{|Q|} |C^q_j| \cdot E_{g \sim \tilde{g}}\mathbb{I}(S \cap R(v, g)) \right] = E_{g \sim \tilde{g}} \left[ \sum_{q \in Q} \sum_{C^q_j \in C_q} a_{q,j} \cdot E_{g \sim \tilde{g}}\mathbb{I}(S \cap R(v, g)) \right].$$

Theorem can be proved.

Let $\tilde{R} = \{\tilde{R}_1, \tilde{R}_2, \cdots, \tilde{R}_\theta\}$ be a collection of random G-RR sets, where in each $\tilde{R}_i \in \tilde{R}$, we denoted it by $\tilde{R}_i = \{R(v_{q,j}, g) : C^q_j \in C_q \text{ and } q \in Q\}$ the $i$-th random G-RR set in collection $\tilde{R}$. Then, according to Eqn. (11) and Eqn. (12) in Theorem 3, the unbiased estimation of our objective function $f(S)$ can be formulated from Eqn. (13) to Eqn. (15), as shown at the bottom of the next page. It is easy to know that the $f(S; \tilde{R})$ is an unbiased estimation function of $f(S)$ under the collection of G-RR sets $\tilde{R}$, and then $\delta(S; \tilde{R})$ and $\phi(S, \tilde{R})$ can be defined in a similar way as shown in Eqn. (14). The $\sigma(V)$ and $\phi(V)$ are constants with $\sigma(V) = n$ and $\phi(V) = \sum_{q \in Q} w_q \sum_{C^q_j \in C_q} a_{q,j}$. Thus, the Eqn. (15) can be further simplified.

**Theorem 4:** Given a collection of G-RR sets $\tilde{R}$, the function $f(S; \tilde{R})$ is monotone and submodular with respect to $S$.

**Proof:** For convenience, we define the marginal gain of $u$ on $S$ as $\Delta_j(u|S; \tilde{R}) = f(S \cup \{u\}; \tilde{R}) - f(S; \tilde{R})$, which can be rewritten as Eqn. (16), shown at the bottom of the next page, based on the expression of Eqn. (15). First, it is monotone because $(S \cup \{u\}) \cap R(v_{q,j}, g^i) = 1$ implies $(S \cup \{u\}) \cap R(v_{q,j}, g^i) = 1$, and thus, we have $\Delta_j(u|S; \tilde{R}) \geq 0$. Next, we show that it is submodular. Given any $S_1 \subseteq S_2 \subseteq V$ with $u \notin S_2$, to show $\Delta_j(u|S_1; \tilde{R}) \geq \Delta_j(u|S_2; \tilde{R})$, it is equivalent to prove $\mathbb{I}(S_1 \cup \{u\}) \cap R(v_{q,j}, g^i) - (S_1 \cap R(v_{q,j}, g^i)) \geq (S_2 \cup \{u\}) \cap R(v_{q,j}, g^i) - (S_2 \cap R(v_{q,j}, g^i))$ according to Eqn. (16). Here, we need to build the connection that we have $\mathbb{I}(S_1 \cup \{u\}) \cap R(v_{q,j}, g^i) = 1$ if $\mathbb{I}(S_2 \cup \{u\}) \cap R(v_{q,j}, g^i) = 1 \wedge \mathbb{I}(S_2 \cap R(v_{q,j}, g^i)) = 1$, which implies $\mathbb{I}(S_2 \cup \{u\}) \cap R(v_{q,j}, g^i) = 1$ and $\mathbb{I}(S_2 \cap R(v_{q,j}, g^i)) = 0$. Obviously, $\mathbb{I}(S_2 \cap R(v_{q,j}, g^i)) = 0$ indicates that $S_2 \cap R(v_{q,j}, g^i) = \emptyset$, naturally we have $S_1 \cap R(v_{q,j}, g^i) = \emptyset$ because of $S_1 \subseteq S_2$. Then, the $\mathbb{I}(S_2 \cup \{u\}) \cap R(v_{q,j}, g^i) = 1$ is enough to infer $\{u\} \cap R(v_{q,j}, g^i) \neq \emptyset$, and thus, we have $\mathbb{I}(S_1 \cup \{u\}) \cap R(v_{q,j}, g^i) = 1$. Thereby, we have $\mathbb{I}(S_1 \cup \{u\}) \cap R(v_{q,j}, g^i) - (S_1 \cap R(v_{q,j}, g^i)) = 1$ and $\Delta_j(u|S_1; \tilde{R}) \geq \Delta_j(u|S_2; \tilde{R})$. $\square$

V. APPROXIMATION ALGORITHM

Based on Theorem 3 and Theorem 4, the original problem can be transformed into a weighted maximum coverage problem, whose objective function $\bar{f}(S; \tilde{R})$ is monotone and submodular given a collection of random G-RR sets $\tilde{R}$. It is much more convenient and efficient than directly solving the original problem. In this section, we first introduce some preliminary knowledge, and then we design our algorithm and conduct the theoretical analysis step by step.

A. Preliminary Analysis

Given a seed set $S$ and a collection of random G-RR sets $\tilde{R}$, we define the generalized coverage as $\Omega(S; \tilde{R}) = \theta \cdot f(S; \tilde{R})$ as shown in Eqn. (15). Given a collection of random G-RR sets $\tilde{R}$, we can apply the MC-Greedy algorithm as shown in Algorithm 1. The algorithm iteratively selects the node $v'_{a_j}$ with the maximum marginal coverage $\Delta_\theta(v'_{a_j}; S_{a_j-1}; \tilde{R}) = \Omega(S_{a_j-1} \cup \{v'_{a_j}; \tilde{R}) - \Omega(S_{a_j-1}; \tilde{R})$ and returns a set $S_a$ as the final solution. Let $S_{a_j}'$ be the solution returned by the MC-Greedy process.
as shown in Algorithm 1, $\tilde{S}_k$ be the optimal size-$k$ set that achieves the maximum weighted coverage $\Omega$, and $S_k^*$ be the optimal solution of the original objective function $f$. The above MC-Greedy algorithm can guarantee
\[
\Omega(S_k^*; \tilde{R}) \geq (1 - 1/e)\Omega(S_k^*; \tilde{R}) \geq (1 - 1/e)\Omega(S_k^*; \tilde{R}),
\]
(17)
because the $\Omega(S; \tilde{R})$ is monotone and submodular with respect to $S$. Then, we have the following concentration bound adapted to the martingale analysis in [21] and [28].

Lemma 2 ([21]): For any $\xi > 0$, given a seed set $S$ and a collection of random G-RR sets $\tilde{R}$, we have
\[
\Pr\left[\Omega(S; \tilde{R}) \leq (1 - \xi)\theta f(S)\right] \leq \exp\left(-\frac{\xi^2\theta f(S)}{2}\right),
\]
(18)
\[
\Pr\left[\Omega(S; \tilde{R}) \geq (1 + \xi)\theta f(S)\right] \leq \exp\left(-\frac{\xi^2\theta f(S)}{2 + \frac{\xi^2}{\theta}}\right).
\]
(19)
Until now, the most excellent algorithm based on the RIS technique to solve the IM problem is OPIM-C [23], where they are optimistic about the selected seed set by the greedy algorithm. Motivated by the idea of OPIM-C, in our CC-DIM problem, we first sample a collection of random G-RR sets $\tilde{R}$ to select a size-$k$ seed set $S_k^*$ in a greedy manner, following Algorithm 1, and derive an upper bound $\bar{f}(S_k^*)$ of $f(S_k^*)$. Second, we sample another collection of random G-RR sets $\tilde{R}_2$ with $|\tilde{R}_2| = |\tilde{R}_1|$ to derive a lower bound $\underline{f}(S_k^*)$ of $f(S_k^*)$.

The algorithm will stop when we have
\[
\frac{\underline{f}(S_k^*)}{\bar{f}(S_k^*)} \geq (1 - 1/e - \epsilon).
\]
(20)
The tighter these bounds are, the fewer the number of random G-RR sets will be, greatly reducing the running time. Based on Lemma 4.2 in [23], we can derive the lower bound $\underline{f}(S_k^*)$ under the $\tilde{R}_2$ with $|\tilde{R}_2| = \theta_2$ as follows:
\[
\underline{f}(S_k^*) = \left[\sqrt{\Omega(S_k^*; \tilde{R}_2)} + \frac{2\eta_2}{\delta} - \frac{\eta_2}{2} - \frac{\eta_2}{18}\right] \cdot \frac{1}{\theta_2}
\]
(21)
where we have $\eta_1 = \ln(1/\delta_1)$ and $\Pr[\underline{f}(S_k^*) > \bar{f}(S_k^*)] \geq 1 - \delta_1$.

In a similar way, we can derive the upper bound $\bar{f}(S_k^*)$ under the $\tilde{R}_1$ with $|\tilde{R}_1| = \theta_1$ as follows:
\[
\bar{f}(S_k^*) = \left(\sqrt{\Omega(S_k^*; \tilde{R}_1)} + \frac{\eta_1}{\theta} + \frac{\eta_1}{2}\right) \cdot \frac{1}{\theta_1},
\]
(22)
where we have $\eta_2 = \ln(1/\delta_2)$ and $\Pr[\bar{f}(S_k^*) < \bar{f}(S_k^*)] \geq 1 - \delta_2$.

Here, the $\Omega(S_k^*; \tilde{R}_1)$ is an upper bound of generalized coverage $\Omega(S_k^*; \tilde{R}_1)$, which satisfies $\Omega(S_k^*; \tilde{R}_1) \leq \Omega(S_k^*; \tilde{R}_1)/(1 - 1/e)$ based on Eqn. (17). To make it tighter, we can construct the upper bound $\Omega(S_k^*; \tilde{R}_1)$ during running the greedy process because of its submodularity. Let $S_n^*$ with $1 \leq a \leq k$ be the set of nodes that are selected in the first $a$ iteration in the MC-Greedy algorithm, and then a tighter upper bound $\Omega(S_n^*; \tilde{R}_1)$ is
\[
\min_{0 \leq a \leq k} \left\{\Omega(S_n^*; \tilde{R}_1) + \sum_{v \in \max MC(S_n^*; k; \tilde{R}_1)} \Delta\Omega(v|S_n^*; \tilde{R}_1)\right\},
\]
where $\max MC(S_n^*; k; \tilde{R}_1)$ is the set of $k$ nodes with the largest marginal coverage in $\tilde{R}_1$ with respect to $S_n^*$.

B. Generalized HIST Algorithm

Given a collection of random G-RR sets $\tilde{R}$, we can quickly obtain a suboptimal solution $S_k^*$ by optimizing function $f(S; \tilde{R})$ through the MC-Greedy algorithm. However, how many random G-RR sets do we need to ensure the approximation ratio as shown in Ineq. (20) is unknown. Thus, in this section, we would like to sample enough random G-RR sets to achieve an accurate estimation of our objective function and guarantee the approximation. Different from the IM problem, to generate a random RR set, it only needs to uniformly sample a node from the graph. However, to generate
Algorithm 2 G-HIST \((G, k, \varepsilon, \delta)\)

1. Initialize: \(\varepsilon_1 = \varepsilon_2 = \varepsilon/2\), \(\delta_1 = \delta_2 = \delta/2\);
2. \(S_k^k = \text{SentinelSet}(G, Q, k, \varepsilon_1, \delta_1)\);
3. \(S_{k-b}^k = \text{RemainingSet}(G, Q, S_b^k, \varepsilon, \varepsilon_2, \delta_2)\);
4. return \(S_b^k \cup S_{k-b}^k\)

a random G-RR set, it needs to uniformly sample a node from every possible community and there are a total of \(\sum_{q \in Q} r_q\) communities. Thus, a specific challenge for sampling in our CC-DIM problem is that the size of a random G-RR set is much larger than that of a random RR set. This not only leads to excessive memory usage but also significantly increases the running time because of the difficult generation and coverage computing process. To overcome this challenge, a feasible strategy is to reduce the average size of a random G-RR set.

Recently, a method has been proposed to reduce memory consumption and running time by reducing the average size of random RR sets in the IM problem, called Hit-and-Stop (HIST) [24], [25]. Motivated by the HIST, our solution to the CC-DIM problem is named Generalized HIST (G-HIST), which can also be divided into the following two stages.

- Sentinel set selection: At this stage, we first generate a small number of random G-RR sets and use it to select a size-\(b\) node set \(S_b^k\) by the MC-Greedy algorithm, which can guarantee \(f(S_b^k) \geq (1 - (1 - 1/k)^b - \varepsilon_1) \cdot f(S_b^k)\) with a high probability.
- Remaining set selection: At this stage, we need to generate enough random G-RR sets to select the remaining size-\((k - b)\) node set \(S_{k-b}^k\). However, in the generation of a random G-RR set \(\tilde{R}\), in each \(R(v_g, g) \in \tilde{R}\), the sampling can be terminated if hitting some node in the sentinel set \(S_b^k\). Therefore, the cost of generating a random G-RR set can be significantly reduced. Then, it returns \(S_b^k \cup S_{k-b}^k\) as the final result and guarantees \(f(S_b^k \cup S_{k-b}^k) \geq (1 - 1/e - \varepsilon_1 - \varepsilon_2) \cdot f(S_b^k)\).

From a high-level perspective, in the stage of sentinel set selection, at the beginning of the MC-Greedy as shown in Algorithm 1, the partial solution \(S_{a-1}^k\) has a small number of nodes, and thus, the value of the marginal gain \(\Delta_G(v|S_{a-1}^k; \tilde{R})\) should be very large. Therefore, the required number of random G-RR sets to select the node with the maximum marginal gain will be small, and it is easy to provide a \((1 - (1 - 1/k)^b - \varepsilon_1)\) approximate solution. With the foundation of the first stage, the sampling and searching process of the second stage can be accelerated. Then, in the stage of remaining set selection, we will need a greater number of random G-RR sets to select nodes in a greedy manner because the value of the marginal gain is relatively small. The average size of random G-RR sets can be significantly pruned based on the partial solution \(S_b^k\) given by the first stage. Thus, the computational cost is reduced without losing the approximation ratio, where the final result can give a \((1 - 1/e - \varepsilon_1 - \varepsilon_2)\) approximation. The G-HIST algorithm can be shown in Algorithm 2. As shown in Algorithm 2, let \(\varepsilon_1 = \varepsilon_2 = \varepsilon/2\) and \(\delta_1 = \delta_2 = \delta/2\), it can return a \((1 - 1/e - \varepsilon)\) approximate solution with at least \(1 - \delta\) probability.

1) Sentinel Set Selection: A natural question is how to determine the size of sentinel set \(S_b^k\). If the size \(b\) is too small, it will reduce the hit rate at the second stage, thus weakening the speed-up effect. If the size \(b\) is too large, this problem will almost be solved, thus worsening the memory consumption and running time. In other words, \(b\) should be carefully determined to balance the cost of sampling at the first stage and the speed-up at the second stage. The process of sentinel set selection is shown in Algorithm 3.

As shown in Algorithm 3, we first take a collection of random G-RR set \(\tilde{R}_1\) and use it to generate a size-\(k\) seed set \(S_b^k\) by the MC-Greedy algorithm as shown in Algorithm 2. In this process, we simultaneously obtain the partial solution \(S_a^k\) with \(1 \leq a \leq k\), which can be applied to compute the upper bound \(\hat{f}(S_a^k)\) by Eqn. (22). However, based on Eqn. (21), we need another collection of random G-RR set \(\tilde{R}_2\), which is independently sampled, to compute the lower bound \(f(S_a^k)\). Let us ignore this point for the time being, where we still apply \(\tilde{R}_1\) to roughly compute the lower bound, denoted by \(\hat{f}(S_a^k)\) to discriminate, for all \(1 \leq a \leq k\). Then, in Line 8 of Algorithm 3, we select the maximum \(a\), denoted by \(b\), such that \(\hat{f}(S_b^k)/\hat{f}(S_a^k) \geq (1 - (1 - 1/k)^a - \varepsilon_1)\). Since the roughly lower bound \(\hat{f}(S_b^k)\) may not be accurate, we generate another collection of random G-RR set \(\tilde{R}_2\) and use it to compute the lower bound \(f(S_b^k)\) by Eqn. (21). Therefore, we can check whether \(S_b^k\) is at least \((1 - (1 - 1/k)^b - \varepsilon_1)\) approximation. If yes, return the \(S_b^k\) directly; if no, make the

Algorithm 3 SentinelSet \((G, Q, k, \varepsilon_1, \delta_1)\)

1. Set \(\theta_1 = 3 \cdot \ln(1/\delta_1)\) and \(\theta_{max}\) according to Eqn. (24);
2. Generate a collection of random G-RR sets \(\tilde{R}_1\) with \(|\tilde{R}_1| = \theta_1\);
3. \(i_{max} \leftarrow \ceil{\log_2(\theta_{max}/\theta)}\);
4. for \(i = 1 \text{ to } i_{max}\) do
5. \(S_b^k \leftarrow \text{MaxCoverage-Greedy}(\tilde{R}_1, k)\);
6. Compute the roughly lower bound \(\hat{f}(S_b^k)\) by Eqn. (21) on \(\tilde{R}_1\) and \(S_b^k\), where \(1 \leq a \leq k\);
7. Get \(\hat{f}(S_a^k)\) by Eqn. (22) on \(\tilde{R}_1\), \(\delta_a = \delta_1/(3i_{max})\);
8. Let \(b\) be the maximum number such that \(\hat{f}(S_b^k)/\hat{f}(S_a^k) \geq (1 - (1 - 1/k)^a - \varepsilon_1)\);
9. Generate a collection of random G-RR sets \(\tilde{R}_2\) with \(|\tilde{R}_2| = |\tilde{R}_1|\) by calling G-RR Set-Sentinel \((G, Q, S_b^k)\) as shown in Algorithm 4;
10. Get \(f(S_b^k)\) by Eqn. (21) on \(\tilde{R}_2\), \(\delta_1 = \delta_1/(6i_{max})\);
11. if \(f(S_b^k)/\hat{f}(S_b^k) \geq (1 - (1 - 1/k)^b - \varepsilon_1)\) then
12. \(\text{return } S_b^k\);
13. Enlarge \(\tilde{R}_2\) until \(|\tilde{R}_2| = 4 \cdot |\tilde{R}_1|\) and recompute \(f(S_b^k)\) by Eqn. (21) on \(\tilde{R}_2\);
14. if \(f(S_b^k)/\hat{f}(S_b^k) \geq (1 - (1 - 1/k)^b - \varepsilon_1)\) then
15. \(\text{return } S_b^k\);
16. Double the size of \(\tilde{R}_1\);
17. \text{return } S_b^k;
lower bound tighter through enlarging \( \tilde{R}_2 \) until \( |\tilde{R}_2| = 4 |R_1| \) and use it to compute the \( f(S_k^*) \) again. If \( S_k^* \) can provide the approximation, return it directly; if not, this implies that this is a high probability that \( S_k^* \) is not a good solution. Thus, we double the collection \( R_1 \) and repeat the above process to reselect a node set until the approximation ratio is satisfied or the maximum number of iterations \( i_{\text{max}} \) is reached.

It is worth noting that in Line 9 of Algorithm 3, the only purpose of the collection of G-RR sets \( \tilde{R}_2 \) is to compute the lower bound \( f(S_k^*) \) for a fixed node set \( S_k^* \). Given a sentinel set \( S_k^* \), the sampling process of a random G-RR set can be optimized, as shown in Algorithm 4. Here, the sampling of collection and searching process of subsequent coverage computation can be significantly improved, which will be widely used in the next stage. As shown in Algorithm 4, we elaborate the process of generating a random G-RR set with the help of sentinel set \( S_k^* \). First, we initialize a map \( \tilde{R} \) (a data structure to represent a G-RR set), where the value of \( \tilde{R}[[q, j]] \) contains a random RR set generated from community \( C_j \), representing the same meaning as \( R[[v_{q,j}, g]] \) in Eqn. (10). For each community \( C_j \), if \( q, j \in Q \), under realization \( g \), we first uniformly select a node \( v_{q,j} \) from \( C_j \). If \( v_{q,j} \) hits the sentinel set \( S_k^* \), we set \( \tilde{R}[[q, j]] \) by \( \Box \) (a placeholder, which means that the RR set \( \tilde{R}[[q, j]] \) has been covered), then terminate the current iteration and enter the next community. If \( v_{q,j} \) does not hit the \( S_k^* \), we add \( v_{q,j} \) into the set \( \tilde{R} \) and queue \( H \) and start a traversal from \( v_{q,j} \) following the reverse direction of its edges in the while loop from Line 13 to Line 23. Here, we use a flag in Line 12 to indicate whether the sampling hits the \( S_k^* \). If yes, the flag will become true, and we set \( \tilde{R}[[q, j]] \) by \( \Box \); if no, the flag will remain false, and we set \( \tilde{R}[[q, j]] \) by sampled RR set \( R \). Given a collection \( \tilde{R}_2 \), when computing the value of \( f(S_k^*) \), we need to compute the value of \( \Omega(S_k^*, \tilde{R}_2) \) as Eqn. (15). Thus, we have \( \Omega(S_k^*, \tilde{R}_2) = 1 \) if and only if \( \tilde{R}_2[[q, j]] = \Box \), which is much easier to compute than before.

Next, how many random G-RR sets are enough in the collection \( R_1 \) to generate a sentinel set \( S_k^* \) with a good approximation? Similar to Lemma 6 in HIST [24], we have the following theorem.

**Theorem 5:** Let \( \tilde{R}_1 \) be the collection of random G-RR sets and \( S_k^* \) be a size-\( b \) node set selected by Algorithm 1 based on \( R_1 \). Given any \( \varepsilon' \) and \( \delta' \), if the size of \( \tilde{R}_1 \) satisfies \( |\tilde{R}_1| \geq 2 \left[ \left( 1 - \left( \frac{1}{k} \right)^b \right) \sqrt{\ln \frac{1}{\varepsilon'}} \left( 1 - \left( \frac{1}{k} \right)^b \right) \left( \ln \left( \frac{n}{k} \right) + \ln 2 \right) \right] ^2, \)

\( \varepsilon'^2 \cdot f(S_k^*) \),

(23)

then we have \( f(S_k^*) \geq (1 - (1 - 1/k)^b - \varepsilon') \cdot f(S_k^*) \) with at least \( 1 - \delta' \) probability.

**Proof:** The proof will be shown in Appendix A. \( \Box \)

According to Theorem 5, we need to give a lower bound of the \( f(S_k^*) \) to get a \( \theta_{\text{max}} \). Here, we define \( f_{\text{min}} = \)

\[ \sup_{S_k \subseteq V} \{ f(S_k) \} = (1 - \lambda) \cdot \frac{k}{n} + \lambda \cdot \min_{q \in Q, C_j \subseteq C_{q_j}} \{ a_{q,j} \} \cdot k \quad \sum_{q \in Q} w_q \sum_{C_j \subseteq C_{q_j}} a_{q,j} \cdot |C_j|, \]

By replacing \( f(S_k^*) \) with \( f_{\text{min}}, \ln \left( \frac{n}{k} \right) \) with \( \ln \left( \frac{n}{k} \right) \), \( 1 - (1/k)^b \) with 1, and setting \( \varepsilon' = \varepsilon_1 \) and \( \delta' = \delta_1/3 \), the maximum number of random G-RR sets in the stage of sentinel set selection is

\[ \theta_{\text{max}} = \frac{2}{\varepsilon_1} \cdot \frac{\ln \frac{\varepsilon}{\delta_1} + \ln \frac{\varepsilon}{\delta_1}}{\varepsilon_1^2 \cdot f_{\text{min}}}. \]

(24)

Thus, if the size of the collection \( \tilde{R}_1 \) is larger than \( \theta_{\text{max}} \), the node set \( S_k^* \) selected based on \( \tilde{R}_1 \) satisfies \( (1 - (1 - 1/k)^b - \varepsilon_1) \cdot f(S_k^*) \), is at least \( \delta_1/3 \). In each of the first \( i_{\text{max}} - 1 \) iterations, the failure probability of the upper bound in Line 7 is \( \delta_1/(3i_{\text{max}}) \) and the failure probabilities of the lower bound in Lines 11 and 14 are \( \delta_1/(6i_{\text{max}}) \) respectively. By the union bound, the total failure probability of the first \( i_{\text{max}} - 1 \) iterations is at most \( 2\delta_1/3 \), and then the sentinel set returned by Algorithm 3 satisfies the desired approximation guarantee with at least \( 1 - \delta_1 \) probability.

2) Remaining Set Selection: After obtaining the sentinel set \( S_k^* \) at the first stage, we make full use of it to accelerate the generation of random G-RR sets and obtain the remaining \( k-b \) seed nodes. The process of remaining set selection is shown in Algorithm 5. Here, we first sample two collections of random RR set \( \tilde{R}_1 \) and \( \tilde{R}_2 \) by invoking Algorithm 4. Based on \( \tilde{R}_1 \), we can select a size-(\( k-b \)) node set \( S_{k-b}^* \) from \( V \setminus S_k^* \) by the MC-Greedy algorithm.

**Remark 1:** In Line 5 of Algorithm 5, given a collection \( \tilde{R}_1 \), we can apply the MC-Greedy algorithm to iteratively select the optimal node. However, there is a difference here since it is a greedy strategy based on \( S_k^* \). Thus, according to Algorithm 1, we make a small change. We initialize \( S_0 = S_k^* \) and iteratively select from \( a = 1 \) to \( k-b \), finally return \( S_{k-b}^* \). Thus, when computing the value of \( \Omega(S_{a-1}, \tilde{R}_1) \) as Eqn. (15) in the MC-Greedy process, we have \( I(S_{a-1} \cap \tilde{R}_1) = 1 \) if and only if \( S_{a-1} \cap \tilde{R}_1 \neq \emptyset \) or \( \tilde{R}_1 \cap \tilde{R}_1 = \emptyset \). This is also the core mystery of our G-HIST algorithm.

After obtaining a feasible solution \( S_k^* \) in Line 6, we use the \( \tilde{R}_1 \) to compute the upper bound \( \tilde{f}(S_k^*) \) and use the \( \tilde{R}_2 \) to compute the lower bound \( f(S_k^*) \). If \( \tilde{f}(S_k^*) \geq \tilde{f}(S_k^*) \) satisfies \( (1 - 1/e - \varepsilon) \) approximation, we return \( S_k^* \) directly; otherwise, we double the collection \( \tilde{R}_1 \) and \( \tilde{R}_2 \) and repeat the above process to reselect the remaining node set until the approximation is satisfied or the maximum number of iterations \( i_{\text{max}} \) is reached.

According to Remark 1, in the stage of remaining set selection, the average size of random G-RR sets can be significantly reduced, and the computational process of coverage in the MC-Greedy algorithm can also be simplified because the RR-set that intersects the sentinel set \( S_k^* \) has been discharged in advanced. Next, how many random G-RR sets are enough in the collection \( \tilde{R}_1 \) to generate a remaining set \( S_{k-b}^* \) with a good approximation guarantee? Similar to Lemma 7 in HIST [24], we can give the following theorem.
Algorithm 4 G-RR Set-Sentinel \((G, Q, S_b^*)\)

1. Initialize \(\tilde{R}\) as a map, and \((q, j)\) is the key for \(q \in Q\) and \(C_q^j \in C_q(G)\);
2. Sample a realization \(g\) from \(G\) randomly;
3. foreach \(q \in Q\) do
   
   foreach \(C_q^j \in C_q(G)\) do
      
      Select a node \(v_{q,j}\) from \(C_q^j\) uniformly;
      
      if \(v_{q,j} \in S_b^*\) then
         
         \(\tilde{R}[(q, j)] \leftarrow \emptyset\);
      
      Continue;
      
      Initialize a set \(R \leftarrow \emptyset\) and a queue \(H \leftarrow \emptyset\);
      
      \(R \leftarrow R \cup \{v_{q,j}\}\);
      
      \(H \leftarrow H \cup \{v_{q,j}\}\); Mark \(v_{q,j}\) as activated;
      
      \(\text{Flag} \leftarrow \text{False};\)
      
      while \(H\) is not empty do
         
         Let \(u\) be the top node of \(H\), pop \(u\) from \(H\);
         
         foreach in-neighbor \(w\) of \(u\) in \(g\) do
            
            if \(w\) is inactivated then
               
               \(\text{Flag} \leftarrow \text{True};\)
               
               \(\text{Break};\)
               
               \(R \leftarrow R \cup \{w\}\);
               
               \(H \leftarrow H \cup \{w\}\); Mark \(w\) as activated;
            
            if \(\text{Flag}\) then
               
               \(\text{Break};\)
            
            else
               
               \(\tilde{R}[(q, j)] \leftarrow R;\)
            
         endforeach
      
   endforeach
   
end foreach

return \(\tilde{R}\);

Algorithm 5 RemainingSet \((G, Q, k, S_b^*, \varepsilon, \varepsilon_2, \delta_2)\)

1. Set \(\theta_1 = 3 \cdot \ln(1/\delta_2)\) and \(\theta_{\text{max}}\) according to Eqn. (24);
2. Generate two collections of random G-RR sets \(\tilde{R}_1\) and \(\tilde{R}_2\) with \(|\tilde{R}_1| = |\tilde{R}_2| = \theta\) by calling G-RR;
3. Set-Sentinel \((G, Q, S_b^*)\) as shown in Algorithm 4;
4. for \(i = 1\) to \(\iota_{\text{max}}\) do
   
   Select a size-\((k-b)\) node set \(S_{k-b}^*\) from \(V \setminus S_b^*\) based on \(\tilde{R}_1\) by the MC-Greedy algorithm;
   
   \(S_k^* \leftarrow S_b^* \cup S_{k-b}^*\);
   
   Get \(f(S_k^*)\) by Eqn. (22) on \(\tilde{R}_1\), \(\delta_u = \delta_2/(3\iota_{\text{max}})\);
   
   Get \(f(S_k^*)\) by Eqn. (21) on \(\tilde{R}_2\), \(\delta_1 = \delta_2/(3\iota_{\text{max}})\);
   
   if \(f(S_k^*)/\mathcal{F}(S_k^*) \geq (1 - 1/e - \varepsilon)\) then
      
      \(\text{return} \ S_{k-b}^*;\)
   
   Double the size of \(\tilde{R}_1\) and \(\tilde{R}_2\) by Algorithm 4;
   
end for

return \(S_{k-b}^*\);

Theorem 6: Given any \(\varepsilon', \delta', \text{and} \ S_b^*\) with \(f(S_b^*) \geq (1 - (1 - 1/k)^b - \varepsilon_1) \cdot f(S_b^*)\), if the size of \(\tilde{R}_1\) satisfies \(\iota_{\text{max}}\) satisfies \(|\tilde{R}_1| \geq 2 \cdot \frac{\sqrt{\ln \frac{3}{\delta_2}} + \sqrt{(1 - \frac{1}{e}) \left(\ln \left(\frac{n-b}{k-b}\right) + \ln \frac{9}{\theta_2}\right)^2}}{\varepsilon^2 \cdot f(S_b^*)}\),

\[ (25) \]

then the remaining set \(S_{k-b}^*\) selected by the adapted MC-Greedy algorithm satisfies \(f(S_k^* \cup S_{k-b}^*) \geq (1 - 1/e - \varepsilon_1 - \varepsilon') \cdot f(S_b^*)\) with at least \(1 - \delta'\) probability.

Proof: The proof will be shown in Appendix B. 

Based on Theorem 6, by replacing \(f(S_b^*)\) with \(f_{\text{min}}\) and setting \(\varepsilon' = \varepsilon_2\) and \(\delta' = \delta_2/3\), the maximum number of random G-RR sets in the stage of remaining set selection is

\[ \theta_{\text{max}} = 2 \cdot \frac{\sqrt{\ln \frac{3}{\theta_2}} + \sqrt{(1 - \frac{1}{e}) \left(\ln \left(\frac{n-b}{k-b}\right) + \ln \frac{9}{\theta_2}\right)^2}}{\varepsilon_2^2 \cdot f_{\text{min}}} . \]

\[ (26) \]

C. Theoretical Results and Complexity

In summary, based on Theorem 5, the sentinel set \(S_b^*\) selected at the first stage satisfies \(f(S_b^*) \geq (1 - (1 - 1/k)^b - \varepsilon_1) \cdot f(S_b^*)\) with at least \(1 - \delta_1\) probability. When it works, based on Theorem 6, the remaining set \(S_{k-b}^*\) selected at the second stage satisfies \(f(S_k^* \cup S_{k-b}^*) \geq (1 - 1/e - \varepsilon_1 - \varepsilon_2) \cdot f(S_b^*)\) with at least \(1 - \delta_2\) probability. As shown in Algorithm 2, by setting \(\varepsilon_1 = \varepsilon_2 = \varepsilon\) and \(\delta_1 = \delta_2 = \delta_2/3\), we have \(f(S_k^* \cup S_{k-b}^*) \geq (1 - 1/e - \varepsilon) \cdot f(S_b^*)\) with at least \(1 - \delta\) probability by the union bound.

For our G-HIST algorithm, the analysis of time complexity is very difficult because the number \(b\) at the first stage cannot be estimated. Thus, we only consider an extreme case, where there is no sentinel set selection stage, namely, \(b = 0\). Then, the Algorithm 5 will directly select a size-\(k\) seed set, whose process is similar to OPIM-C [23]. Thus, when \(\delta < 1/2\), it generates an expected number of \(O(k \ln n + (1/\delta)/(\varepsilon^2 \cdot f(S_b^*)))\) random G-RR sets. To generate a random G-RR set, the worst running time is less than \(O(\sum_{q \in Q} r_q \cdot mn(k \ln n + (1/\delta))/\varepsilon^2 \cdot k)\).

\[ (27) \]

As the search time of the MC-Greedy algorithm is shorter than the generation time, the total time complexity remains unchanged. Now, we can draw the main conclusion.
| Dataset       | n    | m    | Type  | Avg. Degree |
|--------------|------|------|-------|-------------|
| NetScience   | 0.4K | 1.01K| undirected | 5.00        |
| Wiki         | 1.0K | 3.15K| directed | 6.20        |
| HetHEPT      | 12.0K| 118.5K| undirected | 19.8        |
| Epinions     | 75.9K| 508.8K| directed | 13.4        |

Table II: The Datasets Statistics ($K = 10^3$)

Theorem 7 (Main Theorem): The G-HIST as shown in Algorithm 2 can be guaranteed to return a $(1 - 1/e - \varepsilon)$ approximate solution for the CC-DIM problem with at least $1 - \delta$ probability and run in the $\mathcal{O}(\sum_{q \in Q} r_q \cdot mn(k \ln n + \ln(1/\delta))/\varepsilon^2 \cdot k)$ worst expected time.

VI. EXPERIMENTS

In this section, we conduct several experiments on different datasets to validate the effectiveness and efficiency of our G-HIST algorithm for the CC-DIM problem. All of our experiments are programmed in Python and run on a Mac machine. There are four datasets used in the experiments as follows. (1) NetScience [29]: A coauthorship network among scientists to publish papers about network science; (2) Wiki [29]: A who-votes-on-whom network based on the collection of Wikipedia voting; (3) HetHEPT [30]: An academic collaboration relationship on high-energy physics area; and (4) Epinions [30]: A who-trust-whom online social network on Epinions.com, which is a general consumer review site. The statistics of these four datasets are shown in Table II. For the undirected graph, we replace each undirected edge with two reversed directed edges.

A. Experimental Parameter Settings

For the IC model, we use the Weighted Cascade (WC) [8], [31], [32] to set the diffusion probability of each edge. The probability $p_{uv}$ for each edge $(u,v) \in E$ is $1/|N^-(v)|$. As for the parameters in the G-HIST algorithm, we set $\varepsilon = 0.1$ and $\delta = 0.1$. We conduct 1000 Monte Carlo simulations to estimate the objective function given a seed set. Each point in our result is the average over 3 runs.

Because our proposed CC-DIM is a composite community-aware problem, there are multiple community structures in a shared social network. In the objective function as shown in Eqn. (6), we give the $\lambda \in \{0.3, 0.7\}$. Then, the parameter settings are summarized in Table III. Here, we consider two groups of parameter settings, which are denoted by ‘P1’ and ‘P2’ in Table III. For each parameter setting, we then consider three cases of the different numbers of community structures: (1) Case 1: One community structure, denoted by $Q_1 = \{q_1\}$; (2) Case 2: Two community structures, denoted by $Q_2 = \{q_1, q_2\}$; and (3) Case 3: Three community structures, denoted by $Q_3 = \{q_1, q_2, q_3\}$. Here, we have $r_{q_1} = 3$, $r_{q_2} = 4$, and $r_{q_3} = 5$, where the graph will be partitioned into three communities under metric $q_1$, four communities under metric $q_2$, and five communities under metric $q_3$. Their corresponding weights and adjustable coefficients are all listed in Table III.

Taking the parameter setting P1 and case 2 as an example, the weights of metrics $q_1$ and $q_2$ are $w_{q_1} = 0.4$ and $w_{q_2} = 0.6$, where the adjustable coefficient of communities partitioned by metric $q_1$ is $\{a_{q_1,1} = 0.4, a_{q_1,2} = 1, a_{q_1,3} = 1.6\}$ and that partitioned by metric $q_2$ is $\{a_{q_2,1} = 0.4, a_{q_2,2} = 0.8, a_{q_2,3} = 1.2, a_{q_2,4} = 1.6\}$. We can determine the parameters of any case from Table III in a similar way.

In real applications, obtaining a community partition is flexible. We can not only divide the community according to user attributes given by datasets but also use existing algorithms, such as hierarchical clustering [33], modularity maximization [34], [35], and statistical inference methods [36], to partition the graph. In our experiment, to provide valid parameters for comparison, we require that the community partition algorithm returns exact $r$ communities if given an integer $r$ as input. Thus, we use the Normalized Cut [37] as the optimization function for the community partition and adopt the normalized spectral clustering algorithm to solve it in a heuristic way.

Next, we introduce some typical baselines, which will be used for comparison with our G-HIST algorithm.

- **G-HIST**: It is given by Algorithm 2.
- **G-HIST-no-Sentinel**: It directly selects a size-$k$ seed set without the stage of sentinel set selection by invoking Algorithm 5 like RemainingSet $(G, Q, k, \emptyset, \varepsilon, \delta, \delta)$.
TABLE III
THE PARAMETER SETTINGS IN OUR EXPERIMENT

| P  | Case | q  | r_q | u_q | a_q,1 | a_q,2 | a_q,3 | a_q,4 | a_q,5 |
|----|------|----|-----|-----|-------|-------|-------|-------|-------|
| P1 | Q_1  | 3  | 1.0 | 0.4 | 1     | 1.6   | —     | —     | —     |
|    | Q_2  | 3  | 0.4 | 0.4 | 1     | 1.6   | —     | —     | —     |
|    | Q_3  | 4  | 0.6 | 0.4 | 0.8   | 1.2   | 1.6   | —     | —     |
|    | Q_4  | 5  | 0.4 | 0.2 | 0.6   | 1.0   | 1.4   | 1.8   | —     |
| P2 | Q_1  | 3  | 1.0 | 0.1 | 0.1   | 2.8   | —     | —     | —     |
|    | Q_2  | 4  | 0.9 | 0.1 | 0.1   | 0.8   | 3.0   | —     | —     |
|    | Q_3  | 5  | 0.8 | 0.1 | 0.1   | 0.1   | 1.7   | 3.0   | —     |

- **G-IMM**: It uses the IMM [21] to maximize our objective function by setting $\varepsilon = 0.1$ and $\delta = 0.1$.
- **Greedy**: It adopts a greedy hill-climbing algorithm to select the node with maximum marginal gain in each iteration through Monte Carlo simulations.
- **Greedy-IM**: It uses a greedy hill-climbing algorithm to solve the IM problem through Monte Carlo simulations.
- **IMM [21]**: It is a classic sampling-based method of the IM problem by setting $\varepsilon = 0.1$ and $\delta = 0.1$.
- **MaxDegree**: It selects the node with the maximum outdegree in each iteration.
- **Random**: It randomly selects a size-$k$ seed set.

### B. Experimental Results

#### 1) Performance

Fig. 3 and Fig. 4 illustrate the performance comparison achieved by all kinds of algorithms with different parameter settings and $\lambda = 0.7$ under the NetScience and Wiki datasets. In this part, we only consider these two smaller datasets since the Greedy and Greedy-IM algorithms are implemented by Monte Carlo simulations, which cannot be conducted within an acceptable running time for a large network. Here, we make the following observations. First, regardless of which case and parameter settings are used, the performances obtained by the G-HIST, G-HIST-no-Sentinel, and G-IMM algorithms are very close and are obviously better than those of the other baselines. This means that our G-HIST algorithm and preselected sentinel set will not significantly reduce performance, even if the performance fluctuates slightly more than other baselines. Thus, we think that G-HIST, G-HIST-no-Sentinel, and G-IMM are consistent in performance. Second, by comparing G-HIST with Greedy and comparing IMM with Greedy-IM, we can see that the performance of G-HIST (resp. IMM) is slightly better than that of Greedy (resp. Greedy-IM), which means that the methods based on sampling are slightly better than the corresponding methods based on Monte Carlo simulations. In fact, they should be roughly equal. This may be due to the insufficient number of simulations, which leads to an inaccurate estimation of the objective function.

Third, by comparing G-HIST (resp. Greedy) with IMM (resp. Greedy-IM), we find that their performance is roughly similar under P1, where the performance of G-HIST is just slightly better than that of the IMM. However, under P2, the performance of G-HIST is significantly superior to that of the IMM, and the performance gap between G-HIST and IMM will increase as the budget $k$ increases. This is because the weight and coefficient distributions are more uneven under P2, which causes obvious bias. Here, the IMM (Greedy-IM) algorithm that ignores the requirement of diversity will result in an obvious reduction of the objective function, but this effect is not significant under P1 where the weight and coefficient distributions are uniform. Fourth, the performance of G-HIST has a large advantage over the MaxDegree and Random algorithms, and the performance gap between G-HIST and MaxDegree will increase as the budget $k$ increases.

#### 2) Running Time

Table IV shows the running time when $k = 50$ under the parameter setting P2, where we only consider those accurate algorithms for our CC-DIM problem, including G-HIST, G-HIST-no-Sentinel, G-IMM, and Greedy. First, we can see that the computational cost of the simulation-based method, Greedy, is much higher than that of other sampling-based methods. Thus, we do not adopt simulation-based methods in later experiments on larger networks.
datasets. Second, similar to OPIM-C [23], we find a similar trend between G-HIST-no-Sentinel and G-IMM. Under the equivalent setting, the running time of G-HIST-no-Sentinel is approximately 30% to 40% of that of G-IMM. Third, by comparing G-HIST with G-HIST-no-Sentinel, the average size of random G-RR sets at the second stage of G-HIST can be reduced by nearly one order of magnitude, and the running time can also be significantly improved. In our testing, the running time of G-HIST is approximately 60% to 70% of that of G-HIST-no-Sentinel.

3) Different $\lambda$ and Scalability: Fig. 5 and Fig. 6 illustrate the performance comparison achieved by sampling-based algorithms with parameter setting P2 and different $\lambda$ under the HetHEPT and Epinions datasets. In each figure, the left column is when $\lambda = 0.3$ and the right column is when $\lambda = 0.7$. First, the gap in performance between G-IMM and IMM decreases when $\lambda = 0.3$ compared with that when $\lambda = 0.7$. This is because the objective function is close to the influence spread in the IM problem when $\lambda$ is small. Second, in large social networks, our G-HIST has the same advantages in performance and running time as before, further verifying its effectiveness. As we know, the CC-DIM problem will be reduced to the IM problem when $\lambda = 0$. Fig. 7 illustrates the performance comparison achieved by sampling-based algorithms with parameter setting P2 and $\lambda = 0$ under the HetHEPT and Epinions datasets. We can see that G-HIST, G-HIST-no-Sentinel, G-IMM, and IMM all have roughly similar performance, which is in line with our expectations. At this time, our CC-DIM problem is indeed the IM problem. Therefore, the direct use of IM algorithms, IMM, should be the same as other algorithms designed for CC-DIM. By contrast, as shown in Fig. 5 and Fig. 6, the performance obtained by the IMM is obviously worse than other algorithms designed for CC-DIM.

VII. DISCUSSIONS

In this section, we discuss the relationship between our CC-DIM and BIM problem [38]. Then, we give the process of how to extend our sampling-based algorithm to solve general cases in the CC-DIM problem.

### A. Comparison With the BIM Problem

When the utility function is linear, namely $h_{q,j}(x) = a_{q,j} \cdot x$, our CC-DIM problem is a special case of the Benefit Influence Maximization (BIM) defined in [38]. Here, the benefit of each node $u \in V$ can be written as

$$b(u) = \frac{1}{\sigma(V)} + \frac{\lambda}{\phi(V)} \sum_{q \in Q} \sum_{C_j \in C_q} w_q a_{q,j} I(u \in C_j^q) / |C_j^q|.$$  \hspace{1cm} (28)

The BIM problem aims to find a size-$k$ seed set $S$ that can maximize the expected total benefit $\sum_{g \in G} \text{Pr}[g] \cdot \sum_{u \in I_k(g)} b(u)$. By using the Benefit Sampling Algorithm (BSA) proposed in [38], we can directly solve our problem by combining it with any one of the sampling-based approximation algorithms, such as IMM and OPIM-C. This method is more efficient because the sampling process given in BSA is simpler and more efficient than our sampling method proposed in Sec. IV. However, transforming CC-DIM to BIM is only applicable when the utility function is linear. This is not the ultimate goal of this paper. We hope to solve the CC-DIM problem when the utility is an arbitrary concave function.

### B. Extended to General Cases

In general cases, it is challenging to estimate our objective function. To estimate the influence spread in any

| Dataset | Algorithm             | $Q_1$   | $Q_2$   | $Q_3$   |
|---------|-----------------------|---------|---------|---------|
| NetSci  | G-HIST                | 44.5s   | 91.7s   | 2.98m   |
|         | G-HIST-no-Sentinel    | 66.8s   | 2.59m   | 4.72m   |
|         | G-IMM                 | 3.22m   | 7.68m   | 13.3m   |
|         | Greedy                | 51.4m   | 57.2m   | 60.2m   |
| Wiki    | G-HIST                | 2.45m   | 6.14m   | 9.27m   |
|         | G-HIST-no-Sentinel    | 3.73m   | 8.92m   | 14.58m  |
|         | G-IMM                 | 10.2m   | 23.1m   | 41.5m   |
|         | Greedy                | 3.23h   | 3.59h   | 4.00h   |

The running time comparison when $k = 50$ under P2, where $s = \text{second}$, $m = \text{minute}$, and $h = \text{hour}$. Hence, similar to OPIM-C [23], we find a similar trend between G-HIST-no-Sentinel and G-IMM. Under the equivalent setting, the running time of G-HIST-no-Sentinel is approximately 30% to 40% of that of G-IMM. Third, by comparing G-HIST with G-HIST-no-Sentinel, the average size of random G-RR sets at the second stage of G-HIST can be reduced by nearly one order of magnitude, and the running time can also be significantly improved. In our testing, the running time of G-HIST is approximately 60% to 70% of that of G-HIST-no-Sentinel.

3) Different $\lambda$ and Scalability: Fig. 5 and Fig. 6 illustrate the performance comparison achieved by sampling-based algorithms with parameter setting P2 and different $\lambda$ under the HetHEPT and Epinions datasets. In each figure, the left column is when $\lambda = 0.3$ and the right column is when $\lambda = 0.7$. First, the gap in performance between G-IMM and IMM decreases when $\lambda = 0.3$ compared with that when $\lambda = 0.7$. This is because the objective function is close to the influence spread in the IM problem when $\lambda$ is small. Second, in large social networks, our G-HIST has the same advantages in performance and running time as before, further verifying its effectiveness. As we know, the CC-DIM problem will be reduced to the IM problem when $\lambda = 0$. Fig. 7 illustrates the performance comparison achieved by sampling-based algorithms with parameter setting P2 and $\lambda = 0$ under the HetHEPT and Epinions datasets. We can see that G-HIST, G-HIST-no-Sentinel, G-IMM, and IMM all have roughly similar performance, which is in line with our expectations. At this time, our CC-DIM problem is indeed the IM problem. Therefore, the direct use of IM algorithms, IMM, should be the same as other algorithms designed for CC-DIM. By contrast, as shown in Fig. 5 and Fig. 6, the performance obtained by the IMM is obviously worse than other algorithms designed for CC-DIM.
In the previous research on designing IM algorithms, this situation has hardly been considered. Recently, Rui et al. [40] studied a fair IM problem (FiM) whose objective function is the composition of influence spread and polynomial function. The solution to its FiM problem provides us with ideas to solve our problem. By following it, we consider

$$h_{q,j}(x) = a_{q,j} \cdot x^\alpha, 0 < \alpha < 1.$$  \hspace{1cm} (29)

By using Lemma 1 in [40], the composite diversified function as shown in Eqn. (3) can be defined as 

$$\phi(S) = \sum_{q \in Q} \sum_{C^j_q \in C_q} w_q a_{q,j} \left(1 - \alpha \sum_{y=1}^{\infty} \eta(y, \alpha) \left(1 - \gamma(S; C^j_q)\right)^y\right),$$  \hspace{1cm} (30)

where we denote 

$$\eta(y, \alpha) = \begin{cases} 1, & y = 1 \\ (1 - \alpha)(2 - \alpha) \cdots (y - 1 - \alpha), & y \geq 2 \\ y!, & \end{cases}$$

and $$\gamma(S; C^j_q) := \mathbb{E}_{x \sim \mathcal{X}}[|I_g(S) \cap C^j_q|/|C^j_q|]$$ in the subsequent discussion. Let $$X_{q,j}$$ be the random event of whether a randomly selected node in the community $$C^j_q$$ is influenced by a seed set $$S$$. As mentioned above, $$\mathcal{R}$$ is a collection that contains $$\theta$$ random G-RR sets. Let $$X_{q,j}(i)$$ be a random variable for each G-RR set $$R_i \in \mathcal{R}$$, such that $$X_{q,j}(i) = 1$$ if $$R_i(v^1_{q,j}, g^i) \cap S \neq \emptyset$$, and $$X_{q,j}(i) = 0$$ otherwise. Then, we have

$$\mathbb{E}[X_{q,j}] = \gamma(S; C^j_q)$$ and $$\mathbb{E}[X_{q,j}] = 1 - \gamma(S; C^j_q)$$. Based on Eqn. (30) and Lemma 2 in [40], we have

$$\mathbb{E}[X_{q,j}] = 1 - \alpha \sum_{y=1}^{\infty} \eta(y, \alpha)(1 - \mathbb{E}[X_{q,j}])^y = 1 - \alpha \sum_{y=1}^{\infty} \eta(y, \alpha) \frac{(\theta - y)!}{\theta!} \left\{\sum_{i=1}^{\infty} X_{q,j}(i_1) \cdots X_{q,j}(i_y)\right\}.$$  \hspace{1cm} (31)

Next, the unbiased estimator of $$\phi(S)$$ is $$\hat{\phi}(S; \mathcal{R}) = \sum_{q \in Q} \sum_{C^j_q \in C_q} w_q a_{q,j} \left(1 - \alpha \sum_{y=1}^{\infty} \eta(y, \alpha) \frac{1}{\prod_{z=1}^{\infty} \pi_{q,j} - z} \right),$$

where $$\pi_{q,j} = \theta - \sum_{R_i \in \mathcal{R}} X_{q,j}(i)$$. Then, by integrating Eqn. (31) into Eqn. (15), we can get the unbiased estimator of $$f(S)$$. After obtaining the estimation function $$f(S, \mathcal{R})$$, following the same process as shown in Sec. V, we can obtain a $$(1 - 1/\varepsilon)$$ approximate solution with high probability as well.

According to the above discussion, we have solved a large family of utility functions where the utility is polynomial as shown in Eqn. (29). So far, we have not completely solved our CC-DIM problem since we hope that the utility is an arbitrary concave function. For example, if $$h_{q,j}(x) = a_{q,j}^* \cdot \ln(1 + \beta x)$$, then this method cannot be used directly. To this end, we provide the following two feasible solutions.

- Although other forms of concave functions cannot be directly solved, they can be fitted by polynomial functions. For example, if $$h_{q,j}(x) = a_{q,j}^* \cdot \ln(1 + \beta x)$$ is
given, then we will adjust parameters \( a_{q,j} \) and \( \alpha \) such that 
\[ a_{q,j} \cdot x^\alpha \] 
can approach 
\[ a_{q,j} \cdot \ln(1 + \beta x) \] when \( x \in [0, 1] \). 
By replacing 
\[ a_{q,j} \cdot \ln(1 + \beta x) \] with 
\[ a_{q,j} \cdot x^\alpha \], we can deal with it according to the process mentioned above.

- Another solution is to redefine the analytic expression of the objective function. The reason why \( \phi(S) \) can be defined like Eqn. (30) is by incorporating Taylor expansion \((1 + x)^\alpha = 1 + \sum_{y=0}^{\infty} \frac{\alpha^y}{y!} x^y \). Thus, we have 
\[ \phi(S) = \sum_{q \in Q} \sum_{c_j \in c_q} w_q a_{q,j} (1 + (\gamma(S; C_j^q) - 1))^{\alpha} \], and then Eqn. (30) can be obtained. For example, if \( h_{q,j}(x) = a_{q,j} \cdot \ln(1 + \beta x) \) is given, then we will incorporate Taylor expansion \( \ln(1 + \beta x) = \sum_{y=1}^{\infty} \frac{(-1)^{y+1}}{y} (\beta x)^y \) when \( \beta x < 1 \). At this moment, we have
\[
\phi(S) = \sum_{q \in Q} \sum_{c_j \in c_q} w_q a_{q,j} \cdot \ln \left(1 + \beta \gamma(S; C_j^q)\right)
\]
\[
= \sum_{q \in Q} \sum_{c_j \in c_q} w_q a_{q,j} \cdot \left(\sum_{y=1}^{\infty} \frac{(-1)^{y+1}}{y} (\beta \gamma(S; C_j^q))^y\right).
\]

Then, according to similar tricks, we can get an unbiased estimator of \( \phi(S) \) and \( f(S) \) as well.

The second solution looks better because it can accurately solve our CC-DIM problem. Nevertheless, the limitations of the second solution are obvious. First, for each different utility, it needs to derive an analytical expression separately, which is very tedious. Second, the second solution depends on the Taylor expansion of utilities, which requires that a given utility must have a clear Taylor expansion and its Taylor series converges. In fact, this condition is hard to meet, and it is only suitable for the case where the utility function is relatively simple. Looking back at the first solution, polynomial functions have very strong expressive ability, and it is not very difficult to fit a nonlinear function in the \([0, 1]\) interval. Moreover, we provide a complete workflow for it. Therefore, we think the first solution is better, which is more feasible and convenient to be used in real applications.

**VIII. CONCLUSION & FUTURE WORK**

To address the multiplicity of diversity in real social applications, in this paper, we first propose the Composite Community-aware Diversified IM (CC-DIM) problem, which is totally different from the traditional IM problem and Diversified IM problem. Even though its objective function is monotone and submodular, it is extremely difficult to compute. Thus, we create a novel sampling method based on the Generalized Reverse Reachable (G-RR) set to effectively estimate the objective function when the utility is linear and design a two-stage G-HIST algorithm to further improve the memory consumption and time efficiency by significantly reducing the average size of random G-RR sets. According to our theoretical analysis, G-HIST returns a \((1 - 1/e - \epsilon)\) approximate solution with at least \((1 - \delta)\) probability in an acceptable running time. Our experimental results verify our theories and demonstrate the effectiveness and correctness of our proposed algorithm over other state-of-the-art baselines. Finally, we extend our sampling-based method to solve the CC-DIM problem when the utility is an arbitrary monotone and concave function.

When the influence spread is composited with a concave function, such an objective function cannot be effectively estimated by using traditional methods. However, such problems do exist in real IM-based applications. In this paper, we make a preliminary exploration. However, this theoretical design and sampling method are far from optimal, and it is worth further study in the future.

**REFERENCES**

[1] D. Kempe, J. Kleinberg, and É. Tardos, “Maximizing the spread of influence through a social network,” in Proc. 9th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, Aug. 2003, pp. 137–146.
[2] S. Chen, J. Fan, G. Li, J. Feng, K.-L. Tan, and J. Tang, “Online topic-aware influence maximization,” Proc. VLDB Endowment, vol. 8, no. 6, pp. 666–677, Feb. 2015.
[3] S. Tian, S. Mo, L. Wang, and Z. Peng, “Deep reinforcement learning-based approach to tackle topic-aware influence maximization,” Data Sci. Eng., vol. 5, no. 1, pp. 1–11, Mar. 2020.
[4] G. Tong, R. Wang, Z. Dong, and X. Li, “Time-constrained adaptive influence maximization,” IEEE Trans. Computat. Social Syst., vol. 8, no. 1, pp. 33–44, Feb. 2021.
[5] J. Guo and W. Wu, “Adaptive influence maximization: If influential node unwilling to be the seed,” ACM Trans. Knowl. Discovery Data (TKDD), vol. 15, no. 5, pp. 1–23, 2021.
[6] G. Li, S. Chen, J. Feng, K.-L. Tan, and W.-S. Li, “Efficient location-aware influence maximization,” in Proc. ACM SIGMOD Int. Conf. Manag. Data, Jun. 2014, pp. 87–98.
[7] X. Chen, Y. Zhao, G. Liu, R. Sun, X. Zhou, and K. Zheng, “Efficient similarity-aware influence maximization in geo-social network,” IEEE Trans. Knowl. Data Eng., vol. 34, no. 10, pp. 4767–4780, Oct. 2022.
[8] J. Guo, Y. Li, and W. Wu, “Targeted protection maximization in social networks,” IEEE Trans. Netw. Sci. Eng., vol. 7, no. 3, pp. 1645–1655, Jul. 2020.
[9] T. Cai, J. Li, A. Mian, R.-H. Li, T. Sellis, and J. X. Yu, “Target-aware holistic influence maximization in spatial social networks,” IEEE Trans. Knowl. Data Eng., vol. 34, no. 4, pp. 1993–2007, Apr. 2022.
[10] H. Yin, Q. Wang, K. Zheng, Z. Li, J. Yang, and X. Zhou, “Social influence-based group representation learning for group recommendation,” in Proc. IEEE 35th Int. Conf. Data Eng. (ICDE), Apr. 2019, pp. 566–577.
[11] Y. Zhang et al., “Personalized geographical influence modeling for POI recommendation,” IEEE Intell. Syst., vol. 35, no. 5, pp. 18–27, Sep. 2020.
[12] F. Tang, Q. Liu, H. Zhu, E. Chen, and F. Zhu, “Diversified social influence maximization,” in Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining (ASONAM), Aug. 2014, pp. 455–459.
[13] Y. Zhang, “Diversifying seeds and audience in social influence maximization,” in Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining (ASONAM), Aug. 2019, pp. 89–94.
[14] J. Li, T. Cai, K. Deng, X. Wang, T. Sellis, and F. Xia, “Community-diversified influence maximization in social networks,” Inf. Syst., vol. 92, Sep. 2020, Art. no. 101522.
[15] C. Wang, Q. Shi, W. Xian, Y. Feng, and C. Chen, “Efficient diversified influence maximization with adaptive policies,” Knowl.-Based Syst., vol. 213, Feb. 2021, Art. no. 106692.
[16] W. Zhang et al., “GRAIN: Improving data efficiency of graph neural networks via diversified influence maximization,” Proc. VLDB Endowment, vol. 14, no. 11, pp. 2473–2482, Jul. 2021.
[17] W. Chen, C. Wang, and Y. Wang, “Scalable influence maximization for prevalent viral marketing in large-scale social networks,” in Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, Jul. 2010, pp. 1029–1038.
[18] W. Chen, Y. Yuan, and L. Zhang, “Scalable influence maximization in social networks under the linear threshold model,” in Proc. IEEE Int. Conf. Data Eng., Dec. 2010, pp. 88–97.
[19] C. Borgs, M. Brautbar, J. Chayes, and B. Lucier, “Maximizing social influence in nearly optimal time,” in Proc. 24th Annu. ACM-SIAM Symp. Discrete Algorithms, Jan. 2014, pp. 946–957.

[20] Y. Tang, X. Xiao, and Y. Shi, “Influence maximization: Near-optimal time complexity meets practical efficiency,” in Proc. ACM SIGMOD Int. Conf. Manage. Data, Jun. 2014, pp. 75–86.

[21] Y. Tang, Y. Shi, and X. Xiao, “Influence maximization in near-linear time: A martingale approach,” in Proc. ACM SIGMOD Int. Conf. Manage. Data, May 2015, pp. 1539–1554.

[22] H. T. Nguyen, M. T. Thai, and T. N. Dinh, “Stop-and-stare: Optimal sampling algorithms for viral marketing in billion-scale networks,” in Proc. Int. Conf. Manage. Data, Jun. 2016, pp. 695–710.

[23] J. Tang, X. Tang, X. Xiao, and J. Yuan, “Online processing algorithms for influence maximization,” in Proc. Int. Conf. Manage. Data, May 2018, pp. 107–110.

[24] Q. Guo, S. Wang, Z. Wei, and M. Chen, “Influence maximization revisited: Efficient reverse reachable set generation with bound tightened,” in Proc. ACM SIGMOD Int. Conf. Manage. Data, Jun. 2020, pp. 2167–2181.

[25] Q. Guo, S. Wang, Z. Wei, W. Lin, and J. Tang, “Influence maximization revisited: Efficient sampling with bound tightened,” ACM Trans. Database Syst., vol. 47, no. 3, pp. 1–45, Sep. 2022.

[26] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, “An analysis of approximations for maximizing submodular set functions—I,” Math. Program., vol. 14, no. 1, pp. 265–294, 1978.

[27] H. Lin and J. Bilmes, “A class of submodular functions for document summarization,” in Proc. 49th Annu. Meeting Assoc. Comput. Linguistics, Human Lang. Technol., 2011, pp. 510–520.

[28] F. Chung and L. Lu, “Concentration inequalities and martingale inequalities,” in Proc. 34th Ann. ACM Symp. Theory Comput., vol. 3, no. 1, pp. 79–127, Jan. 2006.

[29] R. Rossi and Q. Guo, “The network data repository with interactive graph analytics and visualization,” in Proc. AAAI Conf. Artif. Intell., 2015, vol. 9, no. 1. [Online]. Available: http://networkrepository.com

[30] J. Leskovec and A. Krevl. (Jun. 2014). SNAP Datasets: Stanford Large Network Dataset Collection. [Online]. Available: http://snap.stanford.edu/data

[31] J. Tang, T. Chen, and W. Wu, “A multi-feature diffusion model: Rumor blocking in social networks,” IEEE/ACM Trans. Netw., vol. 29, no. 1, pp. 386–397, Feb. 2021.

[32] J. Guo and W. Wu, “Continuous profit maximization: A study of unconstrained dr-submodular maximization,” IEEE Trans. Computat. Social Syst., vol. 8, no. 3, pp. 768–779, Jun. 2021.

[33] M. Girvan and M. E. J. Newman, “Community structure in social and biological networks,” Proc. Nat. Acad. Sci. USA, vol. 99, no. 12, pp. 7821–7826, Jun. 2002.

[34] M. E. J. Newman, “Fast algorithm for detecting community structure in networks,” Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top., vol. 69, no. 6, Jun. 2004, Art. no. 066133.

[35] M. Chen, K. Kazmin, and B. K. Szymanski, “Community detection via maximization of modularity and its variants,” IEEE Trans. Computat. Social Syst., vol. 1, no. 1, pp. 46–65, Mar. 2014.

[36] B. Karrer and M. E. J. Newman, “Stochastic blockmodels and community structure in networks,” Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top., vol. 83, no. 1, Jan. 2011, Art. no. 016107.

[37] J. Shi and J. Malik, “Normalized cuts and image segmentation,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 22, no. 8, pp. 888–905, Aug. 2000.

[38] H. T. Nguyen, M. T. Thai, and T. N. Dinh, “A billion-scale approximation algorithm for maximizing benefit in viral marketing,” IEEE/ACM Trans. Netw., vol. 25, no. 4, pp. 2419–2429, Aug. 2017.

[39] J. L. W. V. Jensen, “Sur les fonctions convexes et les inégalités entre les valeurs moyennes,” Acta Mathematica, vol. 30, no. 1, pp. 175–193, 1906.

[40] X. Rui, Z. Wang, J. Zhao, L. Sun, and W. Chen, “Scalable fair influence maximization,” 2023, arXiv:2306.00026.