Influence Maximization: Seeding Based on Community Structure

JIANXIONG GUO and WEILI WU, The University of Texas at Dallas

Influence maximization problem attempts to find a small subset of nodes in a social network that makes the expected influence maximized, which has been researched intensively before. Most of the existing literature focus only on maximizing total influence, but it ignores whether the influential distribution is balanced through the network. Even though the total influence is maximized, but gathered in a certain area of social network. Sometimes, this is not advisable. In this article, we propose a novel seeding strategy based on community structure, and formulate the Influence Maximization with Community Budget (IMCB) problem. In this problem, the number of seed nodes in each community is under the cardinality constraint, which can be classified as the problem of monotone submodular maximization under the matroid constraint. To give a satisfactory solution for IMCB problem under the triggering model, we propose the IMCB-Framework, which is inspired by the idea of continuous greedy process and pipage rounding, and derive the best approximation ratio for this problem. In IMCB-Framework, we adopt sampling techniques to overcome the high complexity of continuous greedy. Then, we propose a simplified pipage rounding algorithm, which reduces the complexity of IMCB-Framework further. Finally, we conduct experiments on three real-world datasets to evaluate the correctness and effectiveness of our proposed algorithms, as well as the advantage of IMCB-Framework against classical greedy method.

CCS Concepts: • **Networks → Network algorithms**; • **Theory of computation → Design and analysis of algorithms**;

Additional Key Words and Phrases: Influence maximization, community structure, social network, continuous greedy, matorid, approximation algorithm, IMCB-Framework

ACM Reference format:
Jianxiong Guo and Weili Wu. 2020. Influence Maximization: Seeding Based on Community Structure. *ACM Trans. Knowl. Discov. Data* 14, 6, Article 66 (September 2020), 22 pages.
https://doi.org/10.1145/3399661

1 INTRODUCTION

The online social platforms, such as Facebook, Twitter, LinkedIn, and WeChat, have been growing rapidly over the last years, and have been a major communication platforms. There are more than 1.52 billion users active daily on Facebook and 321 million users active monthly on Twitter. Usually, these social platforms can be represented as online social networks, which is a directed graph, including individuals and their relationship. Viral marketing spreads marketing information

This work is partly supported by the National Science Foundation under grants 1747818 and 1907472.
Authors' addresses: J. Guo and W. Wu, Department of Computer Science, The University of Texas at Dallas, 800 W Campbell Rd, Richardson, Texas, 75080; emails: {jianxiong.guo, weiliwu}@utdallas.edu.

1556-4681/2020/09-ART66 $15.00
https://doi.org/10.1145/3399661

through the word-of-mouth propagations among friends in interpersonal network, which was first proposed by Domingos and Richardson [9, 24]. By giving the most influential users free or coupon samples in social networks, it aims to make the follow-up adoptions maximized. Motivated by the concept of viral marketing, Influence Maximization (IM) appeared as a model to simulate the spread of trust, advertisement, or innovations [18, 19, 32]. The IM problem was formulated by Kempe et al. [18] as a combinatorial optimization problem: selects a subset of nodes with size constraint to make the expected number of follow-up adoptions (influence) maximized. Then, two diffusion models were proposed by them [18], called Independent Cascade model (IC-model) and Linear Threshold model (LT-model), and then, they can be generalized to triggering model. Then, they proved the IM problem is NP-hard and monotone submodular under both IC-model and LT-model. It can be optimized easily by use of existing theory, for example, a simple greedy algorithm can obtain a constant approximation [21]. After this seminal work, a large number of related research studies have been done. They try to overcome the high-time complexity of greedy algorithm [6, 13–15, 20], because greedy algorithm is too slow to apply to large-scale real-word social networks. Later, Reverse Influence Sampling (RIS), proposed by [3], emerged as a mainstream technique to solve IM instead of greedy algorithm [28, 29].

Even if IM has been studied extremely, a majority of the previous research studies focused on studying total IM. This type of model ignores the effect of community structure on influence distribution. In other words, it is possible to appear such a situation that the follow-up adoptions only distribute over partial area of network. Those users in other areas can never have the opportunity to receive information spread from the seed set when budget k is limited. In reality, in some cases, we need to try to avoid this imbalanced influence happening. That is, make the influence distributed more evenly over the network. Let us consider the following possible problems as an example. In the United States, there is a presidential election every 4 years. In order to win a high level of support, each camp will promote and advocate their own presidential candidate. It is not enough to just pursue the maximum total influence, because each state has its own voting rights. Thus, we need to spread the advantage to support our presidential candidate throughout every state. However, in the traditional IM model, some less populated or isolated states, such as Alaska, may be ignored, which is not advisable. Let us take a look at another example. Hunting fugitives down is a very common thing for polices in various countries. The usual method used by national police is to issue a wanted order. Fugitives may be hidden in every corner of this country. Thus, we not only demand the maximum influence of this wanted order, but also require it to be spread to every state, even every county. Same as the previous example, IM model is no longer applicable.

As mentioned above, IM model neglects the distribution of influence over the network. Based on the above analysis, in order to overcome this defect, in this article, we first propose the Influence Maximization with Community budget (IMCB) problem, whose seeding strategy is based on community structure of the networks. IMCB aims to select a subset of nodes, where the number of seed nodes in each community is predefined, such that maximizing the spread of influence. In the above examples of presidential election and wanted order, the states in US correspond to communities in IMCB naturally. In this way, the balance of influence distribution is guaranteed. This community budget can be modeled by use of partition matroid. Naturally, the objective function of IMCB problem is NP-hard and monotone submodular under the triggering model. Then, it can be classified to the problem: monotone submodular maximization under matroid constraint (MSMM). In order to accomplish this goal, we first use classical greedy algorithm, whose theoretical bound is bad and time complexity is high. With the latest research progress, the state-of-the-art algorithm to solve MSMM is continuous greedy (CG) process, proposed by Vondrak [30], which can obtain a $(1 - 1/e)$-approximation. In order to use CG, we need to discretize it first. But even with such algorithm, IMCB problem is still not easy to be solved because the influence cannot be computed

efficiently. The CG assumes that there is a value oracle to compute objective function, unfortunately, it is #P-hard [6] to compute the exact influence. To estimate the expected influence, we adopt the Monte-Carlo simulation usually, but the computational cost is not acceptable. In order to improve its efficiency, randomized strategy based on RIS is popularized gradually [3, 28, 29]. Under the RIS, the value of objective function under the triggering model can be obtained easily. Then, we propose a Sampling-based Discretized-CG, which shows how to solve the IMCB problem by combining CG with RIS. The solution returned by CG is fractional, so we need to round it to integer without losing the approximation ratio. Pipage rounding, first proposed by Ageev et al. [1], can be used to achieve this goal. However, it assumes there is membership oracle for the independent set, which leads to inconvenience to us. Here, we improve pipage rounding algorithm according to some properties of IMCB problem, which simplifies this process, makes it easier to handle, and saves the running time. Finally, IMCB-Framework is formulated, which achieves a $(1 - 1/e)$-approximation with high probability. Our contributions in this article are summarized as follows:

(1) This is the first attempt to study influence maximization based on community structure. Then, IMCB problem is formulated.
(2) We classify IMCB to MSMM problem, and build the solution framework to it. We propose S-Discretized-CG, combining CG with randomized sampling, to overcome #P-hard of computing expected influence and guarantee the theoretical bound.
(3) We propose a simplified version of pipage rounding algorithm without losing its performance.
(4) Our proposed algorithms are evaluated on real-world datasets. The results show that our proposed algorithms are effective and better than greedy and other heuristic algorithms.

Organization: In Section 2, we survey the related work in IM and randomized algorithm. We then present IMCB problem in Section 3, discuss the algorithms and sampling techniques in Section 4, and introduce the simplified pipage rounding algorithm in Section 5. Finally, we conduct experiments and conclude in Sections 6 and 7.

2 RELATED WORK

Domingos and Richardson [9, 24] were the first to study viral marketing, and in [9], they modeled social networks as a Markov random fields and mined from databases. Kempe et al. [18] studied IM as a discrete optimization problem and generalize IC-model and LT-model to triggering model. They derived a $(1 - 1/e)$-approximation by greedy algorithm because of its monotonicity and submodularity. Chen et al. [6, 7] proved it is #P-hard to compute exact influence under this two diffusion model. For community-based IM, Bozorgi et al. [31] proposed a new propagation model based on LT-model under the competitive surrounding, and exploit the community structure to compute the spread of each node locally within its own community. Wang et al. [4] proposed a community-based algorithm to find k influential nodes. They achieved it by first detecting communities and then, selecting communities to find influential nodes by a dynamic programming algorithm.

Becasue of the low efficiency of Monte-Carlo simulation, a lot of researchers attempted to accelerate the process of greedy algorithm by avoiding some unnecessary computations. Leskovec et al. proposed the CELF algorithm [20] with lazy-forward evaluation, which avoids unnecessary computation by estimating the upper bound of influence. CELF++ [11], an improved version of CELF, reduced its time complexity. The effect was not satisfactory until the emergence of RIS. Brogs et al. [3] created RIS firstly, which gave us a new idea to estimate objective function. Based on this idea, a series of efficient randomized algorithms arised like TIM/TIM++ [29], IMM [28] and

Table 1. The Frequently Used Notation Summarization

Notation	Description
$G = (V, E)$	a directed graph G with node set V and edge set E
m, n	the numbder of nodes and edges in G
$N^-(v), N^+(v)$	the incomming and outgoing neighbors set of node v
T_v	triggering set of node v
$g\|\phi$	a realization sampled from triggering distribution ϕ
$\sigma(S\|\phi)$	the influence spread of seed set S under the triggering distribution ϕ
$C(G)$	the community structure of graph G
k_i	the budget of community $C_i \in C(G)$
$\mathcal{M} = (V, \mathcal{I})$	a matorid $\mathcal{M}$ where V is ground set and $\mathcal{I}$ is independent set
$\boldsymbol{x}$	a n-dimensional vector, $\boldsymbol{x} = x_1, \ldots, x_n \in [0, 1]^V$
$F((x))$	the multilinear extension of set function f
$P(\mathcal{M})$	the matroid polytope of $\mathcal{M}$
$\boldsymbol{w}(t)$	the n-dimensional gradient vector of F at time step t
$R, \mathcal{R}$	a Random-RS-T-Set and $\mathcal{R} = \{R_1, R_2, \ldots, R_\theta\}$ is collection of Random-RS-T-Set
$H_{\mathcal{R}}(S\|\phi)$	the coverage fraction of S on $\mathcal{R}$ under the triggering distribution ϕ

SSA/D-SSA [22]. Then, Arora et al. [2] made exhaustive comparisons among existing algorithms for IM by experiments.

CG process appeared first in [30], and mutlilinear extension of a submodular function was introduced by [5]. Besides, they [5] extended CG and pipage rounding to solve MSMM problem with a $(1 - 1/e)$-approximation ratio successfully.

3 PROBLEM FORMULATION

In this section, we talk about influence moded, motivation, and how the problem is formulated. Table 1 summarizes the frequently used notations.

3.1 Influence Model

A social network can be given by a directed graph $G = (V, E)$ where $V = \{v_1, v_2, \ldots, v_n\}$ is the set of n users, and $E = \{e_1, e_2, \ldots, e_m\}$ is the set of m directed edges which describes the relationship between users. The node set and edge set for graph G can be referred as $V(G)$ and $E(G)$, respectively. For an edge $e = (u, v)$, u is an incoming neighbor of v and v is an outgoing neighbor of u. We use $N^-(v)$ and $N^+(v)$ to denote the set of incoming neigbhors and outgoing neighbors of node v, respectively. To simulate the diffusion process, there are two classical diffusion models, IC-model and LT-model, proposed by Kempe et al. [18].

Definition 1 (IC-model). It assumes that diffusion process is executed round by round. In each round, all new activated nodes, for example $u \in V$, have one chance to attempt to activate those inactive nodes v in its outgoing neighbors $N^+(u)$ with probability p_{uv}. Each edge (u, v) is associated with an activation probability $p_{uv} \in [0, 1]$ and the activation process of different edges or different round is independent. Finally, the diffusion process stops if there is no nodes can be activated in future.

Definition 2 (LT-model). It assumes that each edge (u, v) is associated with a weight $b_{uv} \geq 0$ and each node v has a threshold θ_v distributed in $[0, 1]$ uniformly. For each node v, we require that $\Sigma_{u \in N^-(v)} b_{uv} \leq 1$, and define $A(v)$ as the set of active incoming neighbors to node v. The node

v can be activated in this round when satisfying $\Sigma_{u\in A(v)} b_{uv} \geq \theta_v$. Finally, the diffusion process stops if there is no nodes can be activated in future.

IC-model and LT-model are special instances of triggering model [18], where each node v selects a triggering set $T_v \subseteq N^-(v)$ according to some distribution ϕ. For an inactive node v, it can be activated when there exists an incoming neighbor $u \in N^-(v)$ such that $u \in T_v$. Under the IC-model, each incoming neighbor u of v appears in T_v with probabilty p_{uv} independently, and LT-model, node v selects at most one its incoming neighbor from $N^-(v)$, thus, $u \in N^-(v)$ appear in T_v with probabilty b_{uv} exclusively, and $T_v = \emptyset$ with probability $1 - \Sigma_{u\in N^-(v)} b_{uv}$. Now, we can define a realization g of graph G under the triggering model as follows:

Definition 3 (Realization). Under the triggering model, a realization $g|\phi = \{T_{v_1}, T_{v_2}, \ldots, T_{v_n}\}$ is a collection of triggering set for each node in $V(G)$, sampled according to some distribution ϕ. If node u belongs to T_v of node v, the edge (u, v) is live, and otherwise we say it to be blocked.

Let $\Pr[g|\phi]$ be the probability of realization g according to some distribution ϕ, we have:

$$\Pr[g|\phi] = \prod_{i=1}^{n} \Pr[T_{v_i}|\phi] \tag{1}$$

For example, under the IC-model, we have $\Pr[T_v|\phi] = \prod_{u\in T_v} p_{uv} \prod_{u\in N^-(v)\setminus T_v} (1 - p_{uv})$, and LT-model, T_v has at most one node, thus, $\Pr[T_v|\phi] = b_{uv}$ if $T_v = \{u\}$, otherwise $\Pr[T_v|\phi] = 1 - \Sigma_{u\in N^-(v)} b_{uv}$.

Next, the monotonicity and submodularity can be defined here. We say that a set function $f : 2^V \rightarrow \mathbb{R}$ is monotone if for any subsets $S \subseteq T \subseteq V$, $f(S) \leq f(T)$. A set function is submodular if for any $S \subseteq T \subseteq V$ and $u \in V \setminus T$, the marginal gain of u when added to T is less or equal to that when added to S. Formally, $f(S \cup \{u\}) - f(S) \geq f(T \cup \{u\}) - f(T)$.

3.2 Problem Definition

Given a seed set S and a social network $G = (V, E)$, let $\sigma(S|\phi)$ be the expected number of active node (influence) according to triggering distribution ϕ. Thus, we have:

$$\sigma(S|\phi) = \sum_{g\in\mathcal{G}(\phi)} \Pr[g|\phi] \cdot \sigma_g(S) \tag{2}$$

where $\mathcal{G}(\phi)$ is the set of all realizations generated from G according to triggering distribution ϕ and $\sigma_g(\cdot)$ is the number of nodes for which there is a directed path of live edges from a node in S in the realization $g \in \mathcal{G}(\phi)$.

Generally, there exists a community structure given a social network. Community structure is an essential characteristic of social networks. The users can be divided into different groups, and their communication within group are dense, but sparse between groups. Thus, it reveals the internal organizations of social networks. Given network $G = (V, E)$, we assume it exists a unique community structure as $\mathcal{C}(G)$ associated with G, where $\mathcal{C}(G) = \{C_1, C_2, \ldots, C_r\}$ is a partition of $V(G)$. In other words, it means that $V(G) = \bigcup_{i=1}^{r} C_i$, and for any $i, j \in \{1, 2, \ldots, r\}$, we have $C_i \cap C_j = \emptyset$. In this article, we want to find a seed set $S^* \subseteq V(G)$ such that the influence is maximized under a triggering distribution ϕ, but the number of seed nodes $S_i^* \subseteq S^*$ in each community $C_i \in \mathcal{C}(G)$ cannot be more than a predefined threshold k_i, $0 \leq k_i \leq |C_i|$. It is termed as IMCB problem as follows:

Problem 1 (IMCB). *Given a social network $G = (V, E)$, a community structure $\mathcal{C}(G)$ associated with G, a triggering distribution ϕ and a community budget $k = \{k_1, k_2, \ldots, k_r\}$, IMCB aims to select*

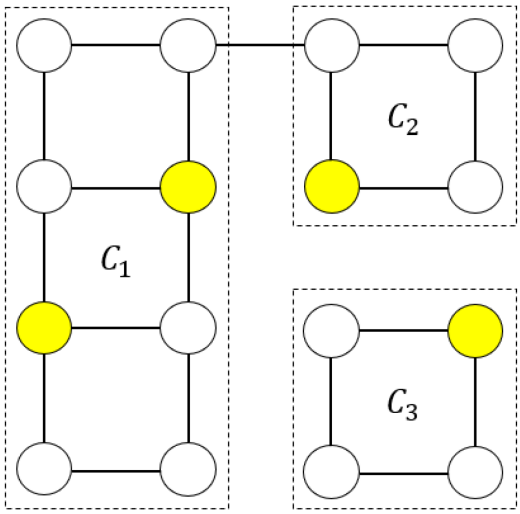


Fig. 1. An example to show how IMCB problem works, the yellow nodes in each community are seed nodes, which is an independent set.

a seed set $S = \bigcup_{i=1}^{r} S_i \subseteq V(G)$ *where* S_i *is the set of seed nodes in community* C_i *and* $|S_i| \leq k_i$*, such that the expected influence* $\sigma(S|\phi)$ *is maximized.*

From above, IMCB problem can be represented as another form. We select a seed set S such that $|S \cap C_i| \leq k_i$ for each $C_i \in \mathcal{C}(G)$, that is,

$$S^* = \arg \max_{S \subseteq V, \forall i: |S \cap C_i| \leq k_i} \sigma(S|\phi) \tag{3}$$

Community budget in IMCB problem can be generalized to the matroid constraint, and we introduce some basic concepts about matroid here. A matroid $\mathcal{M}$ is an order pair $\mathcal{M} = (V, \mathcal{I})$, where V is the ground set, node set in this article, and $\mathcal{I} \subseteq 2^V$ is the collection of independent sets, which satisfies: (1) For all $A \subset B \subseteq V$, if $B \in \mathcal{I}$ then $A \in \mathcal{I}$; and (2) for all $A, B \in \mathcal{I}$ with $|B| > |A|$, we have $\exists v \in B \backslash A$ such that $A \cup \{v\} \in \mathcal{I}$. Under the matroid constraint $\mathcal{M} = (V, \mathcal{I})$, if $S \in \mathcal{I}(\mathcal{M})$, it means that solution $S \subseteq V$ is a feasible solution. The bases of matroid, denoted as $\mathcal{B}$, are those satisfy $\mathcal{B} \in \mathcal{I}$ and $\nexists v \in V$ such that $\mathcal{B} \cup \{v\} \in \mathcal{I}$. All bases of a matroid have the same size. There is a kind of special matroid, Partition Matroid, that is related to our problem:

Definition 4 (Partition Matroid). Given a matroid $\mathcal{M} = (V, \mathcal{I})$, the ground set V is partitioned into disjoint sets $C_1, C_2, \ldots, C_r$, community structure in this article, where $V = \bigcup_{i=1}^{r} C_i$, and let k_i, community budget in this article, be an integer with $0 \leq k_i \leq |C_i|$ for any $i \in \{1, 2, \ldots, r\}$. $\mathcal{M}$ is a partition matorid if $I \in \mathcal{I}(\mathcal{M})$ is equivalent to $|I \cap C_i| \leq k_i$ for any $i \in \{1, 2, \ldots, r\}$.

Thus, in IMCB problem, we need to find $S^* \in \mathcal{I}(M)$ such that $\sigma(S^*|\phi)$ is maximized, and

$$\mathcal{I} = \{S \subseteq V : |S \cap C_i| \leq k_i \text{ for } i = 1, 2, \ldots, r\} \tag{4}$$

Let us take Figure 1 as an example to demonstrate IMCB problem. There are three communities where $\mathcal{C}(G) = \{C_1, C_2, C_3\}$ and $k = \{2, 1, 1\}$. The seeding strategy satisfying Equation (4) may be shown as yellow nodes, which is an independent set.

Theorem 1 ([18]). *The expected influence* $\sigma(S|\phi)$ *is a monotone and submodular function with respect to seed set* S *under the triggering model with distribution* ϕ*.*

Therefore, the IMCB problem can be classified as the problem that maximizing a monotone submodular function subject to a matroid constraint (MSMM).

4 SOLUTION FOR IMCB

From the last section, IMCB problem has been stated, and in this section, we try to give some effective solutions to it, especially sampling techniques. Inspired by recent works, there are some efficient approximation approaches to solve MSMM problem. The state-of-the-art algorithm, CG algorithm, proposed by Calinescu et al. [5], obtained a $(1 - 1/e)$-approximation. Thus, we introduce the basic idea of CG first.

4.1 Preliminary of Continuous Greedy Process

Let $V = \{v_1, v_2, \ldots, v_n\}$ be the ground (node) set and f be a set function such that $f : 2^V \to \mathbb{R}^+$. The multilinear extension of f is defined as follows:

Definition 5 (Multilinear Extension). Let $\boldsymbol{x} \in [0,1]^V$ be a n-dimensional vector, $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$, the multilinear extensiton of f is the function $F : [0,1]^V \to \mathbb{R}^+$ defined as:

$$F(\boldsymbol{x}) = \sum_{S \subseteq V} f(S) \prod_{i \in S} x_i \prod_{i \in V \setminus S} (1 - x_i) \tag{5}$$

Remark 1. There is a probabilistic interpretation of the multilinear extension. Given $\boldsymbol{x} \in [0,1]^V$, we can define S as the random subset of V where each element $i \in V$ is included in S independently with the probability x_i, but not included in S independently with the probability $1 - x_i$. We say $S \sim \boldsymbol{x}$ is the ramdom subset sampled from $\boldsymbol{x}$. Thus, the multilinear extension F of set function f is

$$F(\boldsymbol{x}) = \mathbb{E}_{S \sim \boldsymbol{x}}[f(S)] \tag{6}$$

When f is monotone non-decreasing and submodular, its multilinear extension is non-decreasing and concave along any direction $d \geq 0$. Considering our IMCB problem, $\max_{S \in \mathcal{I}} f(S)$, where $\mathcal{I}$ is defined as Equation (4), it can be relaxed to a multilinear optimization problem as

$$\max\{F(\boldsymbol{x}) : \boldsymbol{x} \in P(\mathcal{M})\} \tag{7}$$

where $P(\mathcal{M})$ is the matroid polytope of matroid $\mathcal{M} = (V, \mathcal{I})$. Then, we can use CG process to obtain a solution of the problem defined in equation (7) with $(1 - 1/e)$-approximation. Given time t, $\boldsymbol{x}(t)$ is a function of t. The CG process can be described as follows:

(1) Initialize $t = 0$ and $\boldsymbol{x}(0) = 0$
(2) Let $v_{max}(\boldsymbol{x}(t)) = \arg\max_{v \in P(\mathcal{M})}(v \cdot \nabla F(\boldsymbol{x}(t))$
(3) $\mathrm{d}\boldsymbol{x}(t)/\mathrm{d}t \leftarrow v_{max}(\boldsymbol{x}(t))$
(4) Output $\boldsymbol{x}(1)$

Remark 2. Considering F is a composite function of time t, $F(\boldsymbol{x}(t))$, we have

$$\frac{\mathrm{d}F(\boldsymbol{x}(t))}{\mathrm{d}t} = \nabla F(\boldsymbol{x}(t)) \cdot \frac{\mathrm{d}\boldsymbol{x}(t)}{\mathrm{d}t} \tag{8}$$

Thus, at step (2), we find a direction $v \in P(\mathcal{M})$ such that maximizing $v \cdot \nabla F(\boldsymbol{x}(t))$ and assign it to $\mathrm{d}\boldsymbol{x}(t)/\mathrm{d}t$. At time t, it makes F grow fastest when $\boldsymbol{x}$ changes along this direction, which is the principle of CG. When time t changes from 0 to 1, we have

$$\boldsymbol{x}(1) = \int_0^1 \frac{\mathrm{d}\boldsymbol{x}(t)}{\mathrm{d}t}\mathrm{d}t = \int_0^1 v_{max}(\boldsymbol{x}(t))\mathrm{d}t \tag{9}$$

Let OPT' be the optimal value of F, adapted from [5], the vector $\boldsymbol{x}(1)$ returned by above CG process satisfies $\boldsymbol{x}(1) \geq (1 - 1/e) \cdot OPT'$.

ALGORITHM 1: Greedy $(G, C(G), k, \phi)$

Input: $G = (V, E)$, community structure $C(G)$, budget set k and triggering distribution ϕ
Output: A seed set S

```
Initialize S ← ∅
Define I as Equation (4)
while true do
    I ← {v|S ∪ {v} ∈ I}
    if I = ∅ then
        break
    end if
    Select v ∈ I that max(σ(S ∪ {v}|φ) − σ(S|φ))
    S ← S ∪ {v}
end while
return seed set S
```

4.2 Algorithm

From above, we have known that the IMCB problem is an instance of monotone submodular maximization problem under matroid constraint, so a greedy algorithm, shown as Algorithm 1, is natural to solve it. This algorithm builds an approximate solution from the empty set, then at each step, it adds a node that gives the maximum marginal influence (according to $\sigma(\cdot)$) but not violating the independence of matroid. For monotone submodular function, the greedy algorithm obtains a $1/(p+1)$-approximation subject to a p-system independence constraint [10]. For the matroid constraint, special case of $p = 1$, the greedy algorithm achieves an approximation ratio of $1/2$ [10]. Even if greedy algorithm is simple and easy to implement, there is a drawback using greedy algorithm to solve IMCB problem: Its approximation ratio is not satisfactory, and in extremely bad cases, it can only reach half of the optimal value.

In Section 4.1, we introduce the CG process, and we show that it can achieve a $(1 - 1/e)$ approximation theoretically. According to Remark 2, Equation (9), we need to compute the value of $\boldsymbol{x}(1)$; however, this integration cannot be computed directly. Assuming $\Delta t = 1/\omega$ and $\omega \in \mathbb{Z}^+$, we have

$$\boldsymbol{x}(1) \approx \sum_{i=0}^{\omega-1} v_{max}(\boldsymbol{x}(i\Delta t)) \cdot \Delta t \tag{10}$$

Based on this discretization strategy, we try our best to make the time scale discretized and balance the granularity of discretization and the error incurred. In time step t, to find $v_{max}(\boldsymbol{x}(t))$, we need to estimate the value of $\nabla F(\boldsymbol{x}(t))$. By Definition (5), we have

$$\frac{\partial F(\boldsymbol{x})}{\partial x_i} = F(\boldsymbol{x}|x_i = 1) - F(\boldsymbol{x}|x_i = 0) \tag{11}$$

Let $X(t)$ be a set that contains each node i with probability $x_i(t)$ at time step t independently, then, each component i of $\nabla F(\boldsymbol{x}(t))$ is equal to $w_i(t) = \mathbb{E}[f(X(t) \cup \{i\}) - f(X(t) \backslash \{i\})]$. Thus, we have $\boldsymbol{w}(t) = \{w_1(t), w_2(t), \ldots, w_n(t)\}$ and $\boldsymbol{w}(t) = \nabla F(\boldsymbol{x}(t))$. Because of the fact that $v_{max}(\boldsymbol{x}(t)) \in P(\mathcal{M})$ and $\boldsymbol{w}(t)$ is non-negative, $v_{max}(\boldsymbol{x}(t))$ corresponds to a base of $\mathcal{M}$. In other words, we find a $I^*(t) \in \mathcal{I}$ such that

$$I^*(t) \in \arg\max_{I(t) \in \mathcal{I}} \left(\boldsymbol{w}(t) \cdot \mathbf{1}_{I(t)} \right) \tag{12}$$

$I^*(t)$ is the maximum-weight independent set at time step t, which can be obtained by hill-climbing strategy, shown as Algorithm 2.

ALGORITHM 2: Max-W-Independent-Set $(G, C(G), k, \boldsymbol{w}(t))$

Input: $G = (V, E)$, community structure $C(G)$, budget set k and weight vector $\boldsymbol{w}(t)$
Output: A independent set I

```
Initialize I ← ∅
Define 𝓘 as Equation (4)
Sort the nodes such that w_1(t) ≥ w_2(t) ≥ ··· ≥ w_n(t)
for i = 1 to n do
    if I ∪ {v_i} ∈ 𝓘 then
        I ← I ∪ {v_i}
    end if
end for
return independent set I
```

Lemma 1 ([27]). *Given a non-negative modular weight function $w : V \to \mathbb{R}^+$, Algorithm 2 leads to a set $I \in \mathcal{I}$ of maximum weight $w(I)$ if and only if $\mathcal{M} = (V, \mathcal{I})$ is a matroid.*

According to Lemma 1, we can know that Algorithm 2 gives us an optimal solution of maximum-weight independent set. In addition, the rank function of matroid $\mathcal{M} = (V, \mathcal{I})$, $\mathcal{I} = \{S \subseteq V : |S \cap C_i| \leq k_i \text{ for } i = 1, 2, \ldots, r\}$, can be defined as

$$r_{\mathcal{M}}(S) = \sum_{i=1}^{r} \min\{|S \cap C_i|, k_i\} \tag{13}$$

Because we have assumed that $k_i \leq |C_i|$ for each $i \in \{1, 2, \ldots, r\}$, we can know $d = r_{\mathcal{M}}(V) = \sum_{i=1}^{r} k_i$. So far, the discretized CG process is formulated, called Discretized-CG. Given graph G, community structure $C(G)$, budget set k, triggering distribution ϕ, time step Δt and sampling number λ, we have

(1) Start with $t = 0$ and $\boldsymbol{x}(0) = \mathbf{0}$.
(2) Obtain $\boldsymbol{w}(t)$: Let $X(t)$ contains each node i independently with probability $x_i(t)$. For each node $i \in V$, we estimate $w_i(t) = \mathbb{E}[\sigma(X(t) \cup \{i\}|\phi) - \sigma(X(t)\backslash\{i\}|\phi)]$ by taking the avergate of λ independent sampling $X(t)$.
(3) $I^*(t) \leftarrow$ Max-W-Independent-Set $(G, C(G), k, \boldsymbol{w}(t))$
(4) $\boldsymbol{x}(t + \Delta t) \leftarrow \boldsymbol{x}(t) + \mathbf{1}_{I^*(t)} \cdot \Delta t$
(5) Increment $t = t + \Delta t$; if $t < 1$, go back to step (2), otherwise, return $\boldsymbol{x}(1)$

Theorem 2. *Let optimal solution $S^* \in \mathcal{I}$, we have $OPT = \sigma(S^*|\phi)$. The fractional solution $\boldsymbol{x}(1)$ returned by Discretized-CG satisfies $F(\boldsymbol{x}(1)) \geq (1 - 1/e) \cdot OPT$ when setting $\Delta t = 1/(40d^2n)$ and $\lambda = 10 \cdot (1 + \log n)/(\Delta t)^2$.*

Proof. This parameter settings are given by [5] for MSMM problem, which can be applied directly here. Thus, we omit the proof. □

Even though Discretized-CG gets a better approximation ratio, it has a common fatal flaw that is the same as the greedy algorithm. They all assume that the objective function $\sigma(\cdot)$ is value oracle. However, things do not turn out the way you want. The computational cost is very high because it is #P-hard [6] to compute the exact value of $\sigma(S|\phi)$ given a seed set S, and the usual computing method is Monte-Carlo simulation. Therefore, it is not advisable to compute $\sigma(\cdot)$ directly, instead of that, we can find an estimator of $\sigma(\cdot)$ by sampling, and then replace $\sigma(\cdot)$ with this estimator in above Discretized-CG process.

4.3 Sampling Techniques

In this subsection, we combine "sampling technique," finding an estimator of $\sigma(S|\phi)$, with discretized-CG process to avoid computing the exact value of $\sigma(S|\phi)$ by Monte-Carlo simulation. The techniques are based on, but different from, RIS [3] for IM problem. The key element of RIS is reversible reachable set (RR-Set), which can be extended to triggering model easily. Here, we refer to RR-Set under the triggering model as RR-T-Set, where T stands for triggering.

Definition 6 (RR-T-Set). Given v be a node in $V(G)$, and g be a realization under triggering distribution ϕ (Definition 3), the RR-T-Set for v in $V(\mathrm{g})$ is the set of nodes in $V(\mathrm{g})$ that exist directed path of live edges in $E(\mathrm{g})$ to v.

The concept of "live" edge is the same as Definition 3, which means that there exists a path such that each edge (u, v) in this path satisfies $u \in T_v$ in g. Given triggering distribution ϕ, a Random-RR-T-Set can be formulated by following process:

(1) Select a node v from V randomly
(2) Generate a realization g of G according to triggering distribution ϕ
(3) Return RR-T-Set given v and g

Let R be a Random-RR-T-Set given triggering distribution ϕ. For a seed set $S \subseteq V$, we can define random variable $z(R, S|\phi) = 1$ if $R \cap S \neq \emptyset$, otherwise $z(R, S|\phi) = 0$. Because R is a Random-RS-T-Set generated by above process given triggering distribution ϕ, $z(R, S|\phi)$ is a random variable. $\mathbb{E}[n \cdot z(R, S|\phi)]$ is an unbiasd estimate of $\sigma(S|\phi)$ [29], thus, we have

$$\mathbb{E}[z(R, S|\phi)] = \frac{\sigma(S|\phi)}{n} \tag{14}$$

Let $\mathcal{R} = \{R_1, R_2, \ldots, R_\theta\}$ be a collection of θ Random-RS-T-Set, which are generated independently given triggering distribution ϕ. Then, we can define $H_{\mathcal{R}}(S|\phi) = \sum_{i=1}^{\theta} z(R_i, S)/\theta$. It is an accurate estimation of $\mathbb{E}[z(R, S|\phi)]$ when the value of θ is large enough. The Chernoff bound is given as

Lemma 2. *Let $Z_i \in [0, 1]$ be θ i.i.d random variables with a mean $\mu = \mathbb{E}[Z_i]$. The Chernoff bound states that for $\delta > 0$,*

$$\Pr\left[\left|\sum Z_i - \mu\theta\right| \geq \delta \cdot \mu\theta\right] \leq 2\exp\left(-\frac{\delta^2}{2+\delta} \cdot \mu\theta\right) \tag{15}$$

Intuitively, we can use some classical randomized frameworks, such as TIM [29] or IMM [28], to solve our IMCB problem. Thus, they can be applied to solve Algorithm 1, which consists of two stages as follows:

(1) Sampling: Generate a collection $\mathcal{R}$ with certain numbers of Random-RR-T-Set according to trigger distribution ϕ independently.
(2) Node selection: Given $\mathcal{R}$ generated in Sampling stages, adpot greedy strategy for maximum coverage to drive a seed set S° such that satisfying matroid constraint.

Here, the objective function is $H_{\mathcal{R}}(\cdot)$ is monotone and submodular. We can obtain $\sigma(S^\circ|\phi) \geq (1/2 - \varepsilon) \cdot OPT$ with high probability by a series of parameter estimation with Chernoff bound. The following conditions should be sufficient enough:

$$\begin{aligned} &n \cdot H_{\mathcal{R}}(S^\circ|\phi) - \sigma(S^\circ|\phi) \leq \varepsilon_1 \cdot OPT \\ &n \cdot H_{\mathcal{R}}(S^*|\phi) \geq (1 - \varepsilon_2) \cdot \mathrm{OPT} \\ &(1 - \varepsilon_2)(1/2) - \varepsilon_1 = 1/2 - \varepsilon \\ &\varepsilon_1, \varepsilon_2 \in (0, 1) \end{aligned} \tag{16}$$

Because it is not the focus of this article and esay to go on, we will not expand in depth here.

However, this framework cannot be applied to solve Discretized-CG directly, because its properties are different from the application scenario of TIM or IMM. The process of Discretized-CG cannot be transformed to a maximum coverage problem by use of unbiased estimator $H_{\mathcal{R}}(\cdot)$. Fortunately, even though that, we can even simplify the sampling process because of the average effect of step (2) in Discretized-CG process.

Remark 3. Here, we only use objective function $\sigma(\cdot)$ to estimate gradient $\boldsymbol{w}(t)$ at time step t. Besides, we estimate gradient $\boldsymbol{w}(t)$ by means of taking average of $\sigma(\cdot|\phi) - \sigma(\cdot|\phi)$. Assuming $\hat{\sigma}(\cdot)$ is an unbiased estimate of $\sigma(\cdot)$, $\hat{\boldsymbol{w}}(t)$ is an unbiased estimate of $\boldsymbol{w}(t)$ as well. Therefore, we can chaim boldly that $\hat{\boldsymbol{w}}(t) = \mathbb{E}[\hat{\sigma}(\cdot) - \hat{\sigma}(\cdot)]$ is almost accurate if the error between $\sigma(\cdot)$ and $\hat{\sigma}(\cdot)$ is restrained within a small certain range.

Thus, relied on above analysis, we can combine the Random-RR-T-Set with Discretized-CG to obtain Sampling-based Discretized-CG, referred as to S-Discretized-CG. Given graph G, community structure $C(G)$, budget set k, triggering distribution ϕ, time step Δt, sampling number λ, parameter ε and N, we have

(1) Generate θ Random-RR-T-Set into $\mathcal{R}$ according to ϕ, where $\theta = n(2n+\varepsilon)(\log N + \log 2)/\varepsilon^2$.
(2) Start with $t = 0$ and $\hat{\boldsymbol{x}}(0) = \boldsymbol{0}$.
(3) Obtain $\hat{\boldsymbol{w}}(t)$: Let $X(t)$ contains each node i independently with probability $\hat{x}_i(t)$. For each node $i \in V$, we estimate $\hat{w}_i(t) = n \cdot \mathbb{E}[H_{\mathcal{R}}(X(t) \cup \{i\}|\phi) - H_{\mathcal{R}}(X(t)\backslash\{i\}|\phi)]$ by taking the avergate of λ independent sampling $X(t)$.
(4) $\hat{I}^*(t) \leftarrow$ Max-W-Independent-Set $(G, C(G), k, \hat{\boldsymbol{w}}(t))$
(5) $\hat{\boldsymbol{x}}(t + \Delta t) \leftarrow \hat{\boldsymbol{x}}(t) + \boldsymbol{1}_{\hat{I}^*(t)} \cdot \Delta t$
(6) Increment $t = t + \Delta t$; if $t < 1$, go back to step (3), otherwise, return $\hat{\boldsymbol{x}}(1)$

Lemma 3. *Given triggering distribution ϕ, if the number of Random-RR-T-Set θ in $\mathcal{R}$ satisfies following condition:*

$$\theta \geq \frac{n(2n+\varepsilon)(\log N + \log 2)}{\varepsilon^2} \tag{17}$$

Then, for any set $S \subseteq V$, the following inequality $|n \cdot H_{\mathcal{R}}(S|\phi) - \sigma(S|\phi)| < \varepsilon$ holds with at least $1 - 1/N$ probability.

Proof. For any set $S \subseteq V$, we define $\mu = \mathbb{E}[z(R, S|\phi)] = \sigma(S|\phi)/n$. Then, we have

$$\begin{aligned} &\Pr\left[|n \cdot H_{\mathcal{R}}(S|\phi) - \sigma(S|\phi)| \geq \varepsilon\right] \\ &= \Pr\left[|\theta \cdot H_{\mathcal{R}}(S|\phi) - \mu\theta| \geq \frac{\varepsilon}{\mu n} \cdot \mu\theta\right] \end{aligned} \tag{18}$$

Let $\delta = \varepsilon/(\mu n)$. By the Chernoff bound, Equation (15), and the fact that $\mu = \sigma(S|\phi)/n \leq 1$, we have

$$\begin{aligned} (18) &\leq 2\exp\left(-\frac{\delta^2}{2+\delta} \cdot \mu\theta\right) \\ &= 2\exp\left(-\frac{\varepsilon^2}{\mu n(2\mu n + \varepsilon)} \cdot \mu\theta\right) \\ &\leq 2\exp\left(-\frac{\varepsilon^2}{n(2n+\varepsilon)} \cdot \theta\right) \\ &\leq \frac{1}{N} \end{aligned}$$

Therefore, $|n \cdot H_{\mathcal{R}}(S|\phi) - \sigma(S|\phi)| < \varepsilon$ holds with at least $1 - 1/N$ probability, the lemma is proved. □

Theorem 3. *Let $X \subseteq V$ be a random variable sampled from the distribution $\boldsymbol{x}$, we have*

$$n \cdot \mathbb{E}[H_{\mathcal{R}}(X|\phi)] = \mathbb{E}[\sigma(X|\phi)] \tag{19}$$

Then, when λ, step (3) of S-Discretized-CG, is large enough, $\hat{\boldsymbol{w}}(t)$ is accurate if the error between $n \cdot H_{\mathcal{R}}(\cdot)$ and $\sigma(\cdot)$ is restrained.

Proof. First, we know that $\mathbb{E}[z(R, X|\phi)] = \sigma(X|\phi)/n$ for any $X \subseteq V$ and Lemma 3, $|n \cdot H_{\mathcal{R}}(X|\phi) - \sigma(X|\phi)| < \varepsilon$ holds with high probability. Then,

$$\mathbb{E}[H_{\mathcal{R}}(X|\phi)] = \sum_{X' \subseteq V} \Pr(X'|\boldsymbol{x}) \cdot H_{\mathcal{R}}(X'|\phi) \tag{20}$$

where $\mathbb{E}[H_{\mathcal{R}}(X'|\phi)] = \sigma(X'|\phi)/n$. The value of $n \cdot H_{\mathcal{R}}(X'|\phi) - \sigma(X'|\phi)$ will be positive or negative with the same probability and same distribution. There are 2^n subsets of V, which is an extremely large number. Shown as Equation (20), it is an accumulation process, meaning that the positive and negative error between $n \cdot H_{\mathcal{R}}(X'|\phi)$ and $\sigma(X'|\phi)$ will offset mutually. Thus, the error between $n \cdot \mathbb{E}[H_{\mathcal{R}}(X'|\phi)]$ and $\mathbb{E}[\sigma(X'|\phi)]$ can be shrinked to a extremely small range. We can think that it is equal generally because the impact on the results is ignorable.

Then, $X \subseteq V$ be a random variable sampled from distribution $\boldsymbol{x}$, and X° is related to X. In step (3) of S-Discretized-CG, $\hat{w}_i(t)$ can be regarded as $\hat{w}_i(t) = n \cdot (\mathbb{E}[H_{\mathcal{R}}(X|\phi)] - \mathbb{E}[H_{\mathcal{R}}(X^\circ|\phi)])$. By Equation (19), $n \cdot \mathbb{E}[H_{\mathcal{R}}(X|\phi)]$ can be replaced with $\mathbb{E}[\sigma(X|\phi)]$. Thus, $\hat{w}_i(t) = w_i(t)$ is established, the theorem is proved. □

In S-Discretized-CG, ε and N are adjustable parameters. When ε is smaller and N is larger, the more accurate the estimate of $n \cdot H_{\mathcal{R}}(X|\phi)$ for $\sigma(X|\phi)$, the more credible the $\hat{w}_i(t) = w_i(t)$ is. Here, we can balance the accuracy and running time by setting different ε and N. So far, we get a fractional vector $\hat{\boldsymbol{x}}(1)$, then we need to round it to an integer vector but cannot violate matroid constraint.

5 SIMPLIFIED PIPAGE ROUNDING

Before, we have assumed that the predefined threshold $k_i \leq |C_i|$ for each community, thus, the fractional vector $\hat{\boldsymbol{x}}$ returned by S-Discretized-CG satisfies $\hat{\boldsymbol{x}}(C_i) = k_i$ for $i = \{1, 2, \ldots, r\}$, where $\hat{\boldsymbol{x}}(C_i) = \sum_{v \in C_i} \hat{\boldsymbol{x}}(v)$. Besides, $\hat{\boldsymbol{x}}(V) = \sum_{i=1}^{r} k_i$ because the result returned by CG is in base polytope $B(\mathcal{M})$. Then, we use randomized pipage rounding [5], which aims to covert $\hat{\boldsymbol{x}}$ into an integral solution that corresponds to a discrete solution $S \in \mathcal{I}$, in other words, to a vextex of $B(\mathcal{M})$. The formal description of randomized pipage rounding is shown in Appendix A. From line 7 to 8 of Algorithm 7, we notice that $\boldsymbol{x}^+ = \boldsymbol{x}^- = \boldsymbol{x}$ may happen due to the fact that there are smaller tight sets containing i or j. Under these circumstances, x_i and x_j cannot be changed, and we need to find a smaller tight set that contains two fractional components. Amazingly, under the IMCB problem, this situation is known in advance. Let us look at an example first.

Example 1. Given social network $G = (V, E)$, we assume there are four nodes in G, thus, $V = \{v_1, v_2, v_3, v_4\}$. There are two communities $\mathcal{C}(G) = \{C_1, C_2\}$ existing in G, Here, $C_1 = \{v_1, v_2\}, k_1 = 1$ and $C_2 = \{v_3, v_4\}$, $k_2 = 1$. Then, we assume the fractional vector $\hat{\boldsymbol{x}}$ returned by S-Discretized-CG is $\hat{\boldsymbol{x}} = (0.6, 0.4, 0.3, 0.7)$, where x_i is corresponding to node v_i. First, considering HitConstraint $(\hat{\boldsymbol{x}}, 2, 3)$, we have $\mathcal{A} = \{\{v_2\}, \{v_1, v_2\}, \{v_2, v_4\}, \{v_1, v_2, v_4\}\}$ and $\delta = \min\{0.6, 0, 0.9, 0.3\} = 0$. Thus, $\hat{\boldsymbol{x}}^+ = \hat{\boldsymbol{x}}$, because $\{v_1, v_2\}$ is a smaller tight set. Similarly, considering HitConstraint $(\hat{\boldsymbol{x}}, 3, 2)$, we have $\delta = 0$ as well because $\{v_3, v_4\}$ is smaller tight set.

ALGORITHM 3: PipageRpund-IMCB $(\mathcal{M}, \boldsymbol{x})$

for each community $C_m \in C$ **do**
  while ($\boldsymbol{x}^m$ is not fractional variables) **do**
    $T \leftarrow C_m$
    while (T contains fractional varibales) **do**
      Pick $i, j \in T$ fractional
      $(\boldsymbol{x}_+^m, A^+) \leftarrow$ Hit-IMC $(\boldsymbol{x}^m, i, j)$
      $(\boldsymbol{x}_-^m, A^-) \leftarrow$ Hit-IMC $(\boldsymbol{x}^m, j, i)$
      $p \leftarrow ||\boldsymbol{x}_+^m - \boldsymbol{x}^m||/||\boldsymbol{x}_+^m - \boldsymbol{x}_-^m||$
      With probability p : $\{\boldsymbol{x}^m \leftarrow \boldsymbol{x}_-^m, T \leftarrow T \cap A^-\}$
      Otherwise : $\{\boldsymbol{x}^m \leftarrow \boldsymbol{x}_+^m, T \leftarrow T \cap A^+\}$
    end while
  end while
  Revise $\boldsymbol{x}$ according to $\boldsymbol{x}^m$
end for
return $\boldsymbol{x}$

ALGORITHM 4: Hit-IMCB $(\boldsymbol{x}^m, i, j)$

Denote $\mathcal{A} = \{A \subseteq C_m | i \in A, j \notin A\}$
Denote $g(A) = r_{\mathcal{M}}(A) - \boldsymbol{x}(A)$
Find $\delta = \min\{g(\{i\}), g(B)\}$ and obtain $A = \{i\}$ or B
$x_i \leftarrow x_i + \delta, x_j \leftarrow x_j - \delta, A' \leftarrow A$
return $(\boldsymbol{x}^m, A')$

From Example 1, we can know that $\delta = 0$ if v_i and v_j belong to different communities when calling HitConstraint $(\boldsymbol{x}, i, j)$. Assuming that $v_i \in C_i$ and $v_j \in C_j$, the A' returned by HitConstraint $(\hat{\boldsymbol{x}}, i, j)$ is C_i because of $\delta = 0$ at this case. Therefore, Algorithm 7 can be simplified based on the community property we have said above, which is shown as Algorithm 3, called PipageRound-IMCB. In Algorithm 3, $\boldsymbol{x}^m$ is a subvector of $\boldsymbol{x}$, where each element in $\boldsymbol{x}^m$ corresponds to a node in community C_m. When $A \subseteq C_m$, we have $\boldsymbol{x}^m(A) = \boldsymbol{x}(A)$. Here, we round the fractional vector community by community, and then merge them together.

Next, in line 2 of Algorithm 6, we need to compute $\delta = \min_{A \in \mathcal{A}}(r_{\mathcal{M}}(A) - \boldsymbol{x}(A))$. We define $g(A) = r_{\mathcal{M}}(A) - \boldsymbol{x}(A)$ where $A \in \mathcal{A}$ and $\mathcal{A} = \{A \subseteq V : i \in A, j \notin A\}$. Since $g(A)$ is a submodular function, there are some existing algorithms [12, 16, 26] with polynomial time to solve this submodular minimization problem. Even that, its running time cannot meet our requirement as well. Do we have the fastest method to compute δ correctly? Let us take a look at the following example.

Example 2. Given social network $G = (V, E)$, we assume there exists a community $C_m \subseteq V$, where $C_m = \{v_1, v_2, v_3, v_4\}$ and $k_m = 2$. Then, we assume the fractional vector $\hat{\boldsymbol{x}}^m$ returned by S-Discretized-CG is $\hat{\boldsymbol{x}}^m = (0.6, 0.2, 0.4, 0.8)$, where x_i is corresponding to node v_i. Considering HitConstraint $(\hat{\boldsymbol{x}}^m, 1, 2)$, we have $\mathcal{A}_1 = \{\{v_1\}, \{v_1, v_3\}, \{v_1, v_4\}, \{v_1, v_3, v_4\}\}$ and $\delta_1 = \min\{0.4, 1, 0.8, 0.2\} = 0.2$. Now, $\hat{\boldsymbol{x}}^m$ changed to $\hat{\boldsymbol{x}}^m = (0.8, 0, 0.4, 0.8)$ and we consider HitConstraint $(\hat{\boldsymbol{x}}^m, 1, 3)$, we have $\mathcal{A}_2 = \{\{v_1\}, \{v_1, v_2\}, \{v_1, v_4\}, \{v_1, v_2, v_4\}\}$ and $\delta_2 = \min\{0.2, 1.2, 0.4, 0.4\} = 0.2$.

From Example 2, we observed an interesting phenomenon that $\delta_1 = \min\{\{v_1\}, \{v_1, v_3, v_4\}\}$ and $\delta_2 = \min\{\{v_1\}, \{v_1, v_2, v_4\}\}$. In general, $g(A)$ shows a trend of increasing first and then decresing with the increase of size of A. Thus, we have following theorem:

ALGORITHM 5: IMCB-Framework

1: Define $\mathcal{M} = (V, \mathcal{I})$
2: Define $\mathcal{I}$ as Equation (4)
3: $\hat{\boldsymbol{x}} \leftarrow$ S-Discretized-CG $(G, C(G), k, \phi, \Delta t, \lambda, \varepsilon, N)$
4: $\boldsymbol{x} \leftarrow$ PipageRpund-IMCB $(\mathcal{M}, \hat{\boldsymbol{x}})$
5: **return** integer vector $\boldsymbol{x}$

Theorem 4. *Given a community $C_m \in \mathcal{C}$, threshold k_m and i, j such that $v_i \in C_m$ and $v_j \in C_m$. Let $\mathcal{A} = \{A \subseteq C_m : i \in A, j \notin A\}$ and $B = C_m \backslash (\{j\} \cup \{a \in C_m : x_k = 0\})$, we have*

$$\min_{A \in \mathcal{A}} g(A) = \min\{g(\{i\}), g(B)\} \tag{21}$$

Proof. The proof is divided into two parts: $|A| \le k_m$ and $|A| > k_m$. When $|A| \le k_m$, let $\mathcal{A}_1 = \{A \in \mathcal{A} : |A| \le k_m\}$, we have $g(\{i\}) \le g(A)$ where $A \in \mathcal{A}_1$. Obviously, for any $A \in \mathcal{A}_1$, $\{i\}$ is the subset of A, thus

$$\begin{aligned} g(A) &= r_{\mathcal{M}}(A) - \boldsymbol{x}(A) \\ &= (r_{\mathcal{M}}(A \backslash \{i\}) + 1) - (\boldsymbol{x}(A \backslash \{i\}) + x_i) \\ &= (r_{\mathcal{M}}(A \backslash \{i\}) - \boldsymbol{x}(A \backslash \{i\})) + (1 - x_i) \\ &= g(A \backslash \{i\}) + g(\{i\}) \\ &\ge g(\{i\}) \end{aligned}$$

where $g(A \backslash \{i\}) \ge 0$. Then, when $|A| > k_m$, let $\mathcal{A}_2 = \{A \in \mathcal{A} : |A| > k_m\}$, we have $g(B) \le g(A)$ where $A \in \mathcal{A}_2$. Here, we denote $B' = C_m \backslash \{j\}$. For any $A \in \mathcal{A}_2$, A is the subset of B', we have

$$\begin{aligned} g(B') &= k_m - \boldsymbol{x}(B') \\ &= k_m - (\boldsymbol{x}(B' \backslash A) + \boldsymbol{x}(A)) \\ &= (k_m - \boldsymbol{x}(A)) + \boldsymbol{x}(B' \backslash A) \\ &= g(A) + \boldsymbol{x}(B' \backslash A) \\ &\le g(A) \end{aligned}$$

where $\boldsymbol{x}(B' \backslash A) \ge 0$. Because $\boldsymbol{x}$ is a fractional vector, $|B| \ge k_m$, and we have $g(B) = k_m - \boldsymbol{x}(B) = g(B') \le g(A)$ where $\boldsymbol{x}(B) = \boldsymbol{x}(B')$. Therefore, $\mathcal{A} = \mathcal{A}_1 \cup \mathcal{A}_2$, for any $A \in \mathcal{A}$, we have $\min_{A \in \mathcal{A}} g(A) = \min\{g(\{i\}), g(B)\}$. □

Therefore, Algorithm 6 can be simplified based on Theorem 4 we have said above, which is shown as Algorithm 4, called Hit-IMCB. Here, we observe that we do not need to check whether $x_j < \delta$ or not, because this case is impossible to occur. $g(B) = k_m - \boldsymbol{x}(B) = x_j$, so $\delta < x_j$ boundly. So far, the eventual algorithm to solve IMCB problem is formulated as Algorithm 5, call IMCB-Framework. The integer vector returned by IMCB-Framework is the indicator vector of the seed set.

6 EXPERIMENT

In this section, we show the effectiveness and efficiency of our proposed algorithms on several real social networks. Our goal is to evaluate Algorithm 5 with some common used baseline algorithms and study the influence distribution over the network.

Table 2. The Statistics of Three Datasets

Dataset	n	m	Type	Average degree
Dataset-1	0.4K	1.01K	directed	4
Dataset-2	1.0K	3.15K	directed	6
Dataset-3	5.2K	14.5K	directed	5

6.1 Dataset Description and Statistics

Our experiments rely on the datasets from networkrepository.com [25], an online network repository. The statistics information of the three datasets is represented in Table 2. The information about these three datasets is shown as follows:

(1) Dataset-1: A co-authorship network, where each edge is a co-authorship among scientists to publish articles in the area of network theory.
(2) Dataset-2: A Wiki network, which is a who-votes-on-whom network collected from Wikipedia.
(3) Dataset-3: A link network, which shows ca-GrQc's link structure by using the interactive network data visualization.

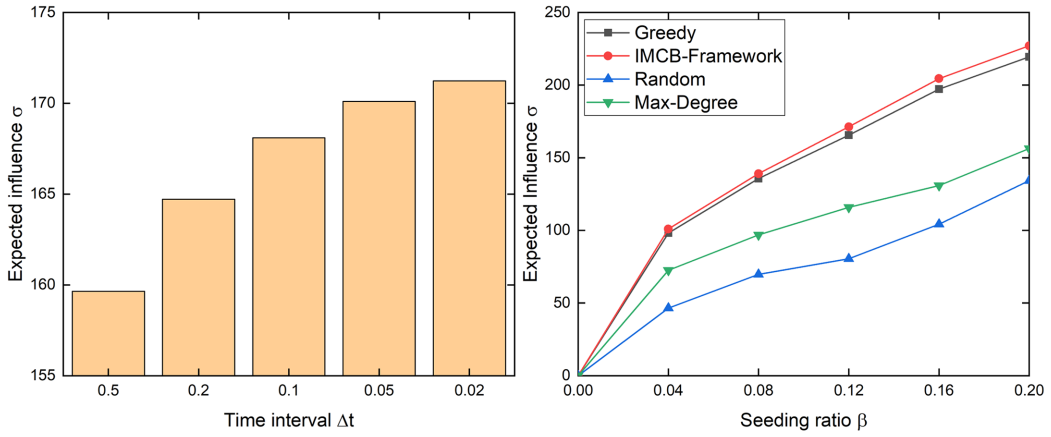
6.2 Experimental Setup

We perform three experiments with different purposes in this section. The first experiment is performed to test whether Δt in Equation (10) is small enough to get a valid approximation. The reason why we need to do this experiment is the $\Delta t = 1/(40d^2n)$ in Theorem 2 is too small, which leads to the running time of S-Discretized-CG unacceptable. We also do not have enough computing power to achieve such precision. After all, whether Monte-Carlo simulation or maximum coverage, they are not value oracle. This is the difference between theoretical model and engineering application. For the perspective of engineering application, we can achieve ideal results but do not need to achieve such precision. Therefore, in first experiment, the value of Δt is ranged in $\Delta t = \{0.5, 0.2, 0.1, 0.05, 0.02\}$, and we can observe the result of different Δt.

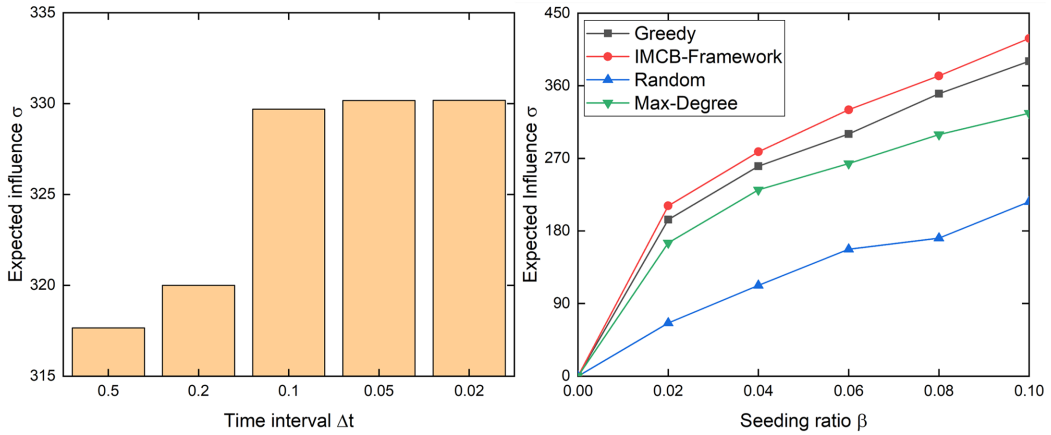
The second expriment is to compare our IMCB-Framework against some common heuristic algorithms to assess the effectiveness of the solution. In this article, our proposed algorithms based on triggering model, here, we choose its two instance, IC-model and LT-model, to demonstrate them. Under the IC-model, the activation probability for each edge $e = (u, v)$ is set as $p_e = 1/|N^-(v)|$. Under the LT-model, the weight for each edge $e = (u, v)$ is set as $b_{uv} = 1/|N^-(v)|$. These settings are widely used in prior work [11, 17, 29]. Then, we compare our proposed algorithm with some common baseline algorithms, which include the following:

(1) *Greedy*: Shown as Algorithm 1, select the node with maximum marginal gain under the matroid constraint $\mathcal{M}$.
(2) *Random*: Select the node randomly from V under the matroid constraint $\mathcal{M}$.
(3) *Max-Degree*: Select the node with the highest out-degree under the matroid constraint $\mathcal{M}$.

For each community C_i, we need to set the budget k_i in advance. Here, we predefine a parameter $\beta \in [0, 1]$, and then, $k_i = \max\{1, \lfloor \beta |C_i| \rfloor\}$ for community C_i. We call β as seeding ratio. In this way, there are at most $\beta|C_i|$ nodes in each community C_i can be selected as the seed nodes. At the same time, we guarantee that there is at least one seed node in each community. Then, in S-Discretized-CG, there are two adjustable parameters ε and N needed to be set, here, $\varepsilon = 0.01 \cdot n$ and $N = 10$. The marginal gain for each node in Greedy algorithm is computed by Monte-Carlo simulation,



(a) Dataset-1



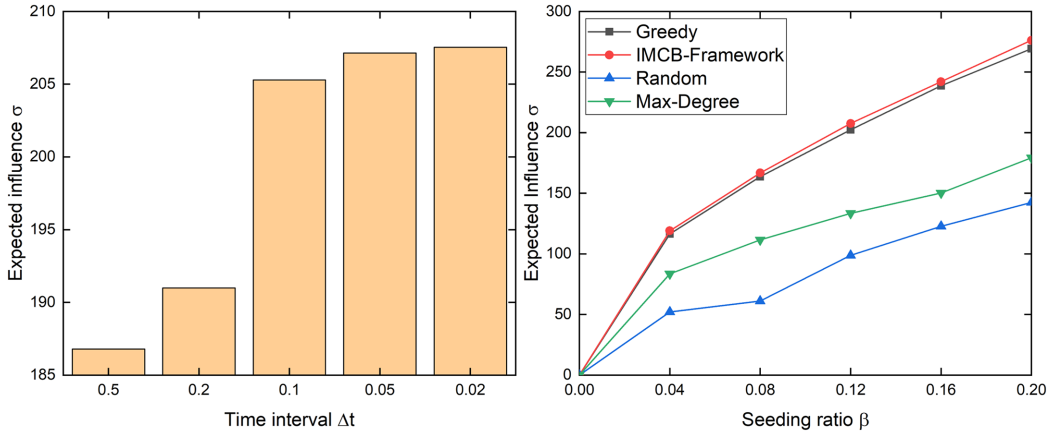
(b) Dataset-2

Fig. 2. The performance changes over seeding ratio β under the IC-model. Left column is achieved by IMCB-Framework with different time interval Δt under (a) $\beta = 0.12$ (b) $\beta = 0.06$; Right column is achieved by different algorithms.

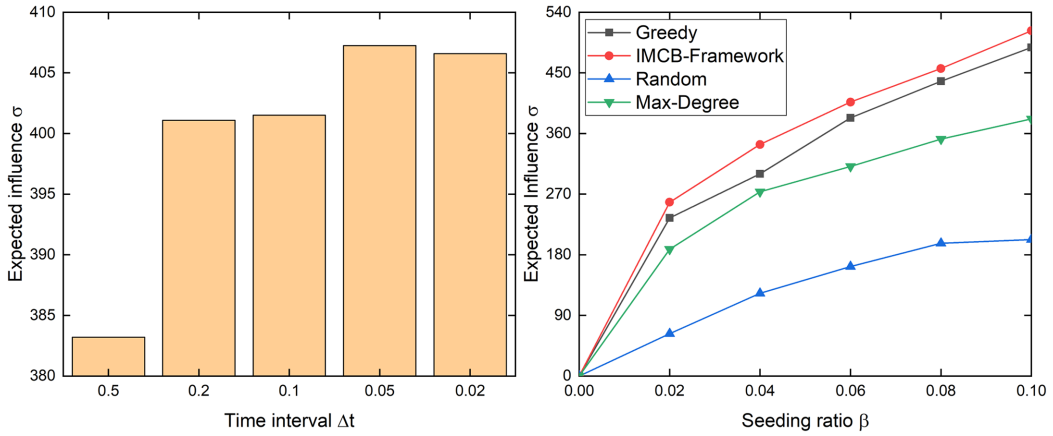
and we estimate the $\sigma(S|\phi)$ by simulating 100 runs. In dataset-1, the value of seeding ratio β is ranged in $\beta = \{0.04, 0.08, 0.12, 0.16, 0.20\}$, and in dataset-2, $\beta = \{0.02, 0.04, 0.06, 0.08, 0.10\}$.

In addition, for a given network, we are required to find its community structure. There exists many known algorithms [23] to find such communities. In our experiment, the classical Clauset–Newman–Moore greedy modularity maximization [8] is used by us to find community structure for a given network. At the beginning, each node is a community, and then we join the pair of communities that makes the modularity increase most until no such pair exists. The community structure and seed information of dataset-1 and dataset-2 is represented in Table 3. Shown as Table 3, the numbers in the first and third row mean the value of seeding ratio β. For example, the value of second row, third column is 23, indicates that the total budget $\sum_{i=1}^{r} k_i$ for seed set is 23 if $\beta = 0.04$ for dataset-1. The column of $|C|$ means the number of community for different datasets.

The third experiment aims to study the the influence distribution over the network. As we know, the expected influence spread of S under the triggering distribution ϕ has been denoted by



(a) Dataset-1



(b) Dataset-2

Fig. 3. The performance changes over seeding ratio β under the LT-model. Left column is achieved by IMCB-Framework with different time interval Δt under (a) $\beta = 0.12$ (b) $\beta = 0.06$; Right column is achieved by different algorithms.

$\sigma(S|\phi)$. Here, we define the expected influence in each community $C_i \in C(G)$ as $\sigma^i(S|\phi)$, thus, we have $\sigma(S|\phi) = \sum_{C_i \in C(G)} \sigma^i(S|\phi)$. In order to quantify the distribution of influence, we define a coverage ratio $\gamma_i(S|\phi) = \sigma^i(S|\phi)/|C_i|$ as a measurement of how many nodes in each community can be activated. Then, the expected coverage ratio for the whole network is $\gamma(S|\phi) = \sum_{C_i \in C(G)} \gamma_i(S|\phi)/|C(G)|$. When the total influence is roughly equal, the higher the expected coverage ratio is, we consider such influence distribution is more balanced.

6.3 Experimental Results

Figures 2 and 3 draws the performance achieved by IMCB-Framework under different time interval Δt and performance comparison with other baseline algorithms. Theoretically, the results of IMCB-Framework should be better and more stable with the decrease of Δt. From the left column of Figures 2 and 3, the effect is not good when $\Delta t = 0.5$, but it has been improved significantly when $\Delta t = 0.2$. When $\Delta t \leq 0.1$, the effect tends to be stable. In a scenario where precision is not very

Table 3. The Community Structure and Seed Information

Dataset-1	$\|C(G)\|$	**0.04**	**0.08**	**0.12**	**0.16**	**0.20**
	19	23	35	47	61	76
Dataset-2	$\|C(G)\|$	**0.02**	**0.04**	**0.06**	**0.08**	**0.10**
	12	25	41	57	74	92

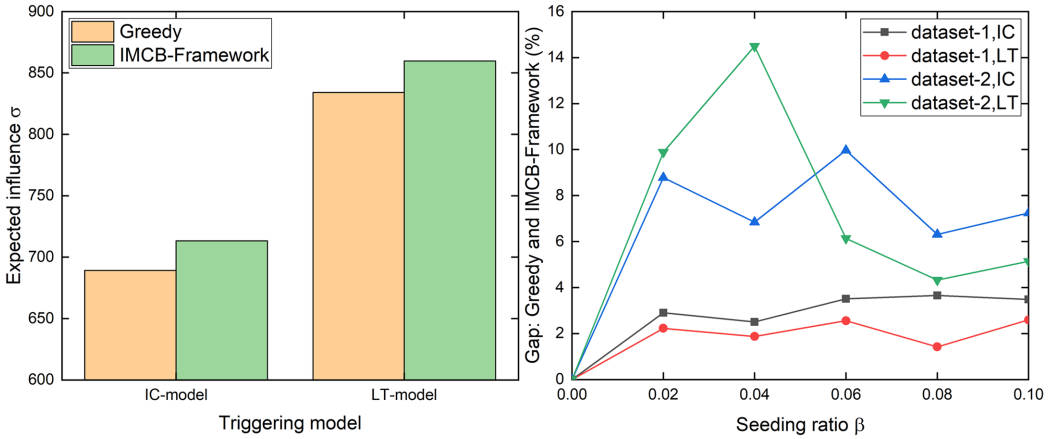


Fig. 4. Left figure is achieved by Greedy and IMCB-Framework with time inteval $\Delta t = 0.1$ and $\beta = 0.005$ under the dataset-3; Right figure is the gap between Greedy and IMCB-Framework with different datasets and triggering models. For dataset-1, horizontal axis needs to be doubled.

important, such as IMCB, $\Delta t = 0.1$ is small enough to obtain a valid result. Therefore, we assume $\Delta t = 0.1$ in subsequent experiments.

According to what we said before, the goal of IMCB problem is to maximize the expected influence but not violating community constraint. Thus, it is valid to set k by use of seeding ratio β, and the larger number of nodes we influence, the better the performance is. The right column of Figures 2 and 3 shows the performance of different algorithms. As depicted of that, the expected influence returned by IMCB-Framework is the largest among these four algorithms, so its performance is the best. In order to characterize the effect of IMCB-Framework on larger network and the gap between Greedy and IMCB-Framework, Figure 4 is provided. From the left figure of Figure 4, under the dataset-3, $\Delta t = 0.1$ and $\beta = 0.005$, IMCB-Framework is better than Greedy as well. The gap, β from 0 to 0.1, between Greedy and IMCB-Framework is shown as right figure of Figure 4. Given β, let S_1 be the seed set returned by Greedy, and S_2 returned by IMCB-Framework, the gap is equal to $|\sigma(S_2|\phi) - \sigma(S_1|\phi)|/\sigma(S_1|\phi) \times 100\%$. The effect from dataset-2 is more obvious than from dataset-1, which may be related to network size and community structure.

According to what we said before, the goal of IMCB problem is to maximize the expected influence but not violating community constraint. Thus, it is valid to set k by use of seeding ratio β, and the larger number of nodes we influence, the better the performance is. The right column of Figures 2 and 3 shows the performance of different algorithms. As depicted of that, the expected influence returned by IMCB-Framework is the largest among these four algorithms, so its performance is the best. In order to characterize the effect of IMCB-Framework on larger network and the gap between Greedy and IMCB-Framework, Figure 4 is provided. From the left figure of Figure 4, under the dataset-3, $\Delta t = 0.1$ and $\beta = 0.005$, IMCB-Framework is better than Greedy as

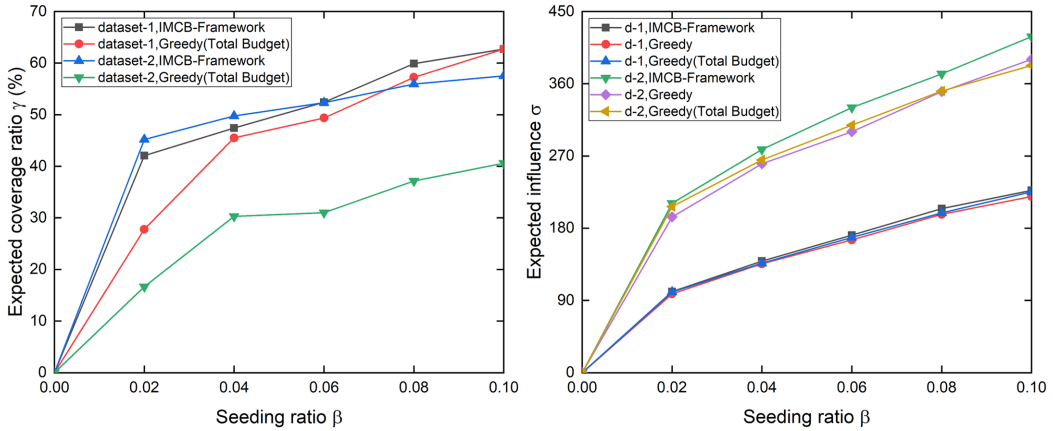


Fig. 5. Left figure is the expected coverage ratio between Greedy(Total Budget) and IMCB-Framework with different datasets under the IC-model; Right figure is achieved by different algorithms under the IC-model to test the performance. For dataset-1, horizontal axis needs to be doubled.

well. The gap, β from 0 to 0.1, between Greedy and IMCB-Framework is shown as right figure of Figure 4. Given β, let S_1 be the seed set returned by Greedy, and S_2 returned by IMCB-Framework, the gap is equal to $|\sigma(S_2|\phi) - \sigma(S_1|\phi)|/\sigma(S_1|\phi) \times 100\%$. The effect from dataset-2 is more obvious than from dataset-1, which may be related to network size and community structure.

Figure 5 draws the expected coverage ratio under the IC-model and performance comparison among IMCB-Framework, Greedy and Greedy (Total Budget). Here, the Greedy is on community budget, but Greedy (Total Budget) is on cardinality budget (equivalent to IM problem). Their correspondence is shown in Table 3, for example, $\beta = 0.04$ on community budget corresponds to cardinality constraint 23 on cardinality budget under the dataset-1. The left figure of Figure 5 shows that the expected coverage ratio of IMCB-Framework is larger than that achieved by Greedy(Total Budget). It means that community budget does help to have a more even distribution of influence. If β is small, this gap will be more obvious. Because there are some communities cannot be influenced completely, this is the last thing we want to see. From Tables 4 and 5, there are some communities whose coverage ratio is very low indeed, even zero, under the total budget when seeding ratio is small. In the Table 4 and Table 5, "com" is community budget, "total" is total budget, and those number of the first row is seeding ratio. The right figure of Figure 5 shows that the expected influence of Greedy(Total Budget) is larger than that of Greedy, but smaller than IMCB-Framework. The overal influence spread will not reduced significantly by community budget, surprisingly, the performance of IMCB-Framework is better than Greedy(Total Budget). Therefore, in most cases, IMCB-Framework is the best chioce for us.

7 CONCLUSION

In this article, we propose IMCB problem to simulate a real scenario where the influence distribution needed to be balanced. IMCB problem can be classified as MSMM problem, which is solved by using IMCB-Framework, and it obtains a $(1 - 1/e)$-approximation with high probability. IMCB-Framework combines CG process with RIS sampling technique to reduce running time, and simplifies the pipage rounding method without losing performance to get a seed set. In experiment, we use two representatives of triggering model, IC and LT, to verify correctness and effectiveness of our proposed algorithms. The major future work is to reduce time complexity further without losing approximation ratio, making it more scalable.

Table 4. Coverage Ratio of All Communities of Dataset-1 Under the IC-model

		0.04		0.08		0.12		0.16		0.20	
com-Id	**#node**	**com**	**total**	**com**	**total**	**com**	**total**	**com**	**total**	**com**	**total**
01	58	0.195	0.149	0.331	0.282	0.438	0.417	0.548	0.570	0.617	0.529
02	50	0.156	0.295	0.291	0.399	0.373	0.383	0.484	0.545	0.562	0.636
03	45	0.213	0.297	0.370	0.368	0.447	0.472	0.573	0.472	0.593	0.613
04	43	0.263	0.263	0.352	0.392	0.475	0.480	0.589	0.517	0.704	0.645
05	27	0.199	0.380	0.377	0.488	0.516	0.480	0.608	0.483	0.588	0.530
06	26	0.215	0.313	0.338	0.315	0.377	0.322	0.766	0.393	0.535	0.602
07	24	0.176	0.174	0.304	0.177	0.426	0.546	0.584	0.547	0.608	0.632
08	22	0.247	0.246	0.318	0.245	0.474	0.402	0.593	0.538	0.541	0.501
09	17	0.240	0.243	0.244	**0.000**	0.365	0.242	0.465	0.494	0.464	0.562
10	11	0.362	0.026	0.369	0.374	0.364	0.514	0.530	0.602	0.600	0.733
11	09	0.531	0.521	0.513	0.516	0.515	0.510	0.513	0.512	0.625	0.518
12	09	0.384	0.392	0.387	0.389	0.393	0.393	0.383	0.386	0.464	0.386
13	08	0.316	**0.000**	0.315	0.217	0.321	0.317	0.319	0.322	0.527	0.516
14	07	0.680	0.681	0.687	0.683	0.690	0.677	0.677	0.686	0.689	0.689
15	06	0.660	**0.000**	0.664	0.661	0.662	0.662	0.657	0.661	0.659	0.662
16	06	0.604	0.587	0.591	0.595	0.590	0.596	0.594	0.600	0.591	0.598
17	05	0.717	0.712	0.718	0.716	0.712	0.714	0.713	0.715	0.713	0.714
18	03	0.912	**0.000**	0.912	0.919	0.905	0.921	0.918	0.920	0.914	0.919
19	03	0.919	**0.000**	0.927	0.908	0.912	0.333	0.926	0.917	0.921	0.926

Table 5. Coverage Ratio of All Communities of Dataset-2 Under the IC-model

		0.02		0.04		0.06		0.08		0.10	
com-Id	**#node**	**com**	**total**	**com**	**total**	**com**	**total**	**com**	**total**	**com**	**total**
01	297	0.194	0.261	0.285	0.286	0.346	0.353	0.403	0.387	0.453	0.442
02	231	0.257	0.278	0.333	0.348	0.402	0.352	0.448	0.392	0.506	0.440
03	154	0.241	0.255	0.319	0.320	0.377	0.414	0.416	0.461	0.476	0.484
04	141	0.199	0.107	0.274	0.266	0.335	0.226	0.364	0.313	0.423	0.317
05	023	0.349	0.120	0.298	0.119	0.313	0.453	0.478	0.474	0.497	0.474
06	016	0.385	0.385	0.390	0.305	0.392	0.542	0.391	0.540	0.467	0.540
07	008	0.505	**0.000**	0.498	**0.000**	0.505	0.566	0.571	0.567	0.380	0.571
08	006	0.399	0.048	0.390	0.082	0.393	0.117	0.417	0.605	0.439	0.602
09	005	0.332	0.062	0.361	0.062	0.394	0.060	0.396	0.065	0.389	0.068
10	003	0.805	0.012	0.821	0.019	0.823	0.107	0.828	0.110	0.870	0.117
11	003	0.757	0.474	1.000	0.830	1.000	0.532	1.000	0.543	1.000	0.816
12	002	1.000	**0.000**	1.000	**0.000**	1.000	0.000	1.000	0.000	1.000	0.000

APPENDIX

A PIPAGE ROUNDING ALGORITHM

The formal description of pipage rounding algorithm under matroid constraint is shown as Algorithms 6 and 7 [5].

ALGORITHM 6: HitConstraint $(\boldsymbol{x}, i, j)$

Denote $\mathcal{A} = \{A \subseteq V | i \in A, j \notin A\}$
Find $\delta = \min_{A \in \mathcal{A}}(r_{\mathcal{M}}(A) - \boldsymbol{x}(A))$ and obtain $A \in \mathcal{A}$
if $x_j < \delta$ **then**
  $x_i \leftarrow x_i + x_j, x_j \leftarrow 0, A' \leftarrow \{j\}$
else
  $x_i \leftarrow x_i + \delta, x_j \leftarrow x_j - \delta, A' \leftarrow A$
end if
return $(\boldsymbol{x}, A')$

ALGORITHM 7: PipageRound $(\mathcal{M}, \boldsymbol{x})$

while ($\boldsymbol{x}$ is not fractional variables) **do**
  $T \leftarrow V$
  while (T contains fractional varibales) **do**
    Pick $i, j \in T$ fractional
    $(\boldsymbol{x}^+, A^+) \leftarrow$ HitConstraint $(\boldsymbol{x}, i, j)$
    $(\boldsymbol{x}^-, A^-) \leftarrow$ HitConstraint $(\boldsymbol{x}, j, i)$
    if $(\boldsymbol{x}^+ = \boldsymbol{x}^- = \boldsymbol{x})$ **then**
      $T \leftarrow T \cap A^+$
    else
      $p \leftarrow ||\boldsymbol{x}^+ - \boldsymbol{x}||/||\boldsymbol{x}^+ - \boldsymbol{x}^-||$
      With probability p : $\{\boldsymbol{x} \leftarrow \boldsymbol{x}^-, T \leftarrow T \cap A^-\}$
      Otherwise : $\{\boldsymbol{x} \leftarrow \boldsymbol{x}^+, T \leftarrow T \cap A^+\}$
    end if
  end while
end while
return $\boldsymbol{x}$

REFERENCES

[1] Alexander A. Ageev and Maxim I. Sviridenko. 2004. Pipage rounding: A new method of constructing algorithms with proven performance guarantee. *Journal of Combinatorial Optimization* 8, 3 (2004), 307–328.

[2] Akhil Arora, Sainyam Galhotra, and Sayan Ranu. 2017. Debunking the myths of influence maximization: An in-depth benchmarking study. In *Proceedings of the 2017 ACM International Conference on Management of Data*. ACM, 651–666.

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

[4] Arastoo Bozorgi, Saeed Samet, Johan Kwisthout, and Todd Wareham. 2017. Community-based influence maximization in social networks under a competitive linear threshold model. *Knowledge-Based Systems* 134 (2017), 149–158.

[5] Gruia Calinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. 2011. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM Journal on Computing* 40, 6 (2011), 1740–1766.

[6] Wei Chen, Chi Wang, and Yajun Wang. 2010. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1029–1038.

[7] Wei Chen, Yifei Yuan, and Li Zhang. 2010. Scalable influence maximization in social networks under the linear threshold model. In *Proceedings of the 2010 IEEE International Conference on Data Mining*. IEEE, 88–97.

[8] Aaron Clauset, Mark E. J. Newman, and Cristopher Moore. 2004. Finding community structure in very large networks. *Physical Review E* 70, 6 (2004), 066111.

[9] Pedro Domingos and Matt Richardson. 2001. Mining the network value of customers. In *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 57–66.
[10] Marshall L. Fisher, George L. Nemhauser, and Laurence A. Wolsey. 1978. An analysis of approximations for maximizing submodular set functionsâII. In *Polyhedral Combinatorics*. Springer, 73–87.
[11] Amit Goyal, Wei Lu, and Laks V. S. Lakshmanan. 2011. Celf++: optimizing the greedy algorithm for influence maximization in social networks. In *Proceedings of the 20th International Conference Companion on World Wide Web*. ACM, 47–48.
[12] Martin Grötschel, László Lovász, and Alexander Schrijver. 2012. *Geometric Algorithms and Combinatorial Optimization*. Vol. 2. Springer Science & Business Media.
[13] Jianxiong Guo, Tiantian Chen, and Weili Wu. 2020. Budgeted coupon advertisement problem: Algorithm and robust analysis. *IEEE Transactions on Network Science and Engineering* (2020), 1–1. DOI: https://doi.org/10.1109/TNSE.2020.2964882
[14] Jianxiong Guo, Yi Li, and Weili Wu. 2019. Targeted protection maximization in social networks. *IEEE Transactions on Network Science and Engineering* (2019), 1–1. DOI: https://doi.org/10.1109/TNSE.2019.2944108
[15] Jianxiong Guo and Weili Wu. 2019. A Novel scene of viral marketing for complementary products. *IEEE Transactions on Computational Social Systems* 6, 4 (2019), 797–808.
[16] Satoru Iwata, Lisa Fleischer, and Satoru Fujishige. 2001. A combinatorial strongly polynomial algorithm for minimizing submodular functions. *Journal of the ACM* 48, 4 (2001), 761–777.
[17] Kyomin Jung, Wei Chen, and Wooram Heo. 2011. *IRIE: A Scalable Influence Maximization Algorithm for Independent Cascade Model and Its Extensions*. Technical Report.
[18] David Kempe, Jon Kleinberg, and Éva Tardos. 2003. Maximizing the spread of influence through a social network. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 137–146.
[19] Paul Lagrée, Olivier Cappé, Bogdan Cautis, and Silviu Maniu. 2018. Algorithms for online influencer marketing. *ACM Transactions on Knowledge Discovery from Data* 13, 1 (2018), 3.
[20] Jure Leskovec, Andreas Krause, Carlos Guestrin, Christos Faloutsos, Jeanne VanBriesen, and Natalie Glance. 2007. Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 420–429.
[21] George L. Nemhauser, Laurence A. Wolsey, and Marshall L. Fisher. 1978. An analysis of approximations for maximizing submodular set functionsâI. *Mathematical Programming* 14, 1 (1978), 265–294.
[22] Hung T. Nguyen, My T. Thai, and Thang N. Dinh. 2016. Stop-and-stare: Optimal sampling algorithms for viral marketing in billion-scale networks. In *Proceedings of the 2016 International Conference on Management of Data*. ACM, 695–710.
[23] Maryam Ramezani, Ali Khodadadi, and Hamid R. Rabiee. 2018. Community detection using diffusion information. *ACM Transactions on Knowledge Discovery from Data* 12, 2 (2018), 20.
[24] Matthew Richardson and Pedro Domingos. 2002. Mining knowledge-sharing sites for viral marketing. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 61–70.
[25] Ryan A. Rossi and Nesreen K. Ahmed. 2015. The network data repository with interactive graph analytics and visualization. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*. Retrieved from http://networkrepository.com.
[26] Alexander Schrijver. 2000. A combinatorial algorithm minimizing submodular functions in strongly polynomial time. *Journal of Combinatorial Theory, Series B* 80, 2 (2000), 346–355.
[27] Alexander Schrijver. 2003. *Combinatorial Optimization: Polyhedra and Efficiency*. Vol. 24. Springer Science & Business Media.
[28] Youze Tang, Yanchen Shi, and Xiaokui Xiao. 2015. Influence maximization in near-linear time: A martingale approach. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*. ACM, 1539–1554.
[29] Youze Tang, Xiaokui Xiao, and Yanchen Shi. 2014. Influence maximization: Near-optimal time complexity meets practical efficiency. In *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*. ACM, 75–86.
[30] Jan Vondrák. 2008. Optimal approximation for the submodular welfare problem in the value oracle model. In *Proceedings of the 14th Annual ACM Symposium on Theory of Computing*. ACM, 67–74.
[31] 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 *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1039–1048.
[32] Ruidong Yan, Yi Li, Weili Wu, Deying Li, and Yongcai Wang. 2019. Rumor blocking through Online link deletion on social networks. *ACM Transactions on Knowledge Discovery from Data* 13, 2 (2019), 16.

Received October 2019; revised February 2020; accepted May 2020