

Article

Influence Maximization under Fairness Budget Distribution in Online Social Networks

Bich-Ngan T. Nguyen ^{1,2} , Phuong N. H. Pham ^{1,2} , Van-Vang Le ^{3,*}  and Václav Snášel ² 

¹ Faculty of Information Technology, Ho Chi Minh City University of Food Industry, 140 Le Trong Tan Street, Ho Chi Minh City 700000, Vietnam

² Department of Computer Science, Faculty of Electrical Engineering and Computer Science, VŠB-Technical University of Ostrava, 17.listopadu 15/2172, 708 33 Ostrava, Czech Republic

³ Faculty of Information Technology, Ton Duc Thang University, Ho Chi Minh City 700000, Vietnam

* Correspondence: levanvang@tdtu.edu.vn

Abstract: In social influence analysis, viral marketing, and other fields, the influence maximization problem is a fundamental one with critical applications and has attracted many researchers in the last decades. This problem asks to find a k -size seed set with the largest expected influence spread size. Our paper studies the problem of fairness budget distribution in influence maximization, aiming to find a seed set of size k fairly disseminated in target communities. Each community has certain lower and upper bounded budgets, and the number of each community's elements is selected into a seed set holding these bounds. Nevertheless, resolving this problem encounters two main challenges: strongly influential seed sets might not adhere to the fairness constraint, and it is an NP-hard problem. To address these shortcomings, we propose three algorithms (FBIM1, FBIM2, and FBIM3). These algorithms combine an improved greedy strategy for selecting seeds to ensure maximum coverage with the fairness constraints by generating sampling through a Reverse Influence Sampling framework. Our algorithms provide a $(1/2 - \epsilon)$ -approximation of the optimal solution, and require $O\left(kT \log\left((8 + 2\epsilon)n \frac{\ln \frac{2}{\delta} + \ln \binom{n}{k}}{\epsilon^2}\right)\right)$, $O\left(kT \log \frac{n}{\epsilon^2 k}\right)$, and $O\left(\frac{T}{\epsilon} \log \frac{k}{\epsilon} \log \frac{n}{\epsilon^2 k}\right)$ complexity, respectively. We conducted experiments on real social networks. The result shows that our proposed algorithms are highly scalable while satisfying theoretical assurances, and that the coverage ratios with respect to the target communities are larger than those of the state-of-the-art alternatives; there are even cases in which our algorithms reaches 100% coverage with respect to target communities. In addition, our algorithms are feasible and effective even in cases involving big data; in particular, the results of the algorithms guarantee fairness constraints.

Keywords: fairness budget; submodular maximization; influence maximization; sampling; approximation algorithm; viral marketing

MSC: 68W25; 68D15; 91B99; 90B60



Citation: Nguyen, B.-N.T.; Pham, P.N.H.; Le, V.-V.; Snášel, V. Influence Maximization under Fairness Budget Distribution in Online Social Networks. *Mathematics* **2022**, *10*, 4185. <https://doi.org/10.3390/math10224185>

Academic Editors: Gaogao Dong, Jianguo Liu and Zhao Kang

Received: 16 August 2022

Accepted: 2 November 2022

Published: 9 November 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In the digital information age, use of online social networks (OSNs) has become indispensable and widespread for many people. Current OSNs have millions or even billions of users, such as Facebook, Twitter, Instagram, LinkedIn, Youtube, and others. OSNs can rapidly influence people by sharing behavior, content, or messages between one person and another [1]. This propagation is similar to the way viruses spread exponentially. Based on this powerful feature of OSNs, brands or organizations use an online marketing tactic through OSNs known as viral marketing. Because this tactic can rapidly spread information, effectively promote products, usefully support candidates in elections, etc., on a large scale, it often helps to achieve high results with modest investment costs [2]. However, an effective viral marketing campaign must seed content with groups of influential people in OSNs.

The process of identifying a group of such individuals is referred to as the Influence Maximization (IM) problem [3].

Influence Maximization problem. Given a ground set V , which is the set of users in the social network, and an information diffusion model, let $k \in \mathbb{Z}^+$ be a global cardinality constraint, $S \subseteq V$ and $|S| \leq k$ be the set of influential users that needs to be determined, and $f(S)$ be the influence function measuring the expected number of users in V that can be affected by members in S under the given information diffusion model. The above problem finds that

$$\max_{S \subseteq V, |S| \leq k} f(S) \quad (1)$$

IM is a celebrated topic that has attracted the interest of many experts during the last decade. Although Kempe et al. first introduced this problem under the name *influence maximization* in 2003 [3], Richardson et al. in 2002 were the first to study the problem as one of maximizing an advertiser's earnings on a social network [4]. IM is crucial in a wide variety of applications, including viral marketing [5,6], social network analysis [7,8], social problems such as financial inclusion [9], HIV prevention for homeless youth [10], propagation of information for disease spread [11], and other issues.

There are many efficient approaches to this problem that assess the spread of influence: the formulation of the discrete optimization problem, which is NP-hard [3]; using continuous-time models [12]; ranking and score-based heuristics [13]; or an excited new approach using sketches and the *Reverse Influence Sampling* (RIS) framework, as suggested by Borgs et al. [14]. Numerous publications have used RIS to solve the IM problem, with positive results [6,15–17].

However, most of these existing solutions to the problem IM focus on the most influential nodes to maximize the total number of affected nodes. This means that these methods only aim to find the most influential users to maximize the number of affected people. Such approaches are not interest in whether the influenced people are fairly distributed over the network. Thus, there is a high probability that users in groups that do not contain seeded users will not be influenced or will not receive the information. On the contrary, it is these users that must be affected.

For example, viral marketing is a significant and standard application of the IM problem. The objective is to maximize the profits of advertisers by promoting products to users on OSNs [4]. OSNs are fertile ground for the advertising field; they have millions or billions of users, and the number of users continuously increases every day. However, it is troubling that users of OSNs often encounter advertising content that they have viewed too many times or already purchased, and thus do not care about. In contrast, users who are the right customers for these ads do not receive them. This is boring and frustrating for users. Therefore, the challenge is how to spread advertising content to the right customers and other potential customer groups in OSN communities.

In recent years, to conquer the above weakness of IM, a new variant of it has been developed that has attracted the attention of researchers. This is the problem of *fair influence maximization* (FIM), which aims to ensure a fair distribution for the groups in the final set of selected nodes [18,19]. In other words, it guarantees coverage propagation in the target communities. However, each of the existing methods offers a unique perspective on fairness. To the best of our knowledge, there are no publications that have considered both minimum and maximum budget constraints for each group to guarantee equitable distribution.

Fueled by this challenge, in this paper we study FIM under budget threshold constraint, setting upper and lower bounded budgets to choose seeds in each community in order to guarantee fairness (termed FBIM; see Definition 5). Specifically, our contributions are as follows.

- We propose three algorithms to solve the FBIM problem. These algorithms provide a $(1/2 - \epsilon)$ -approximation of the optimal solution that works efficiently with big data.

(1) The first algorithm is FBIM1, which uses a combination of several methods. First, it generates sampling through an RIS framework with the *dynamic stop-and-stare algorithm*, known as DSSA [6], and adds a fairness constraint during seed set selection. Our algorithm has $O\left(kT \log\left((8 + 2\epsilon)n^{\frac{\ln \frac{2}{\epsilon} + \ln(\frac{n}{k})}{\epsilon^2}}\right)\right)$ complexity. The results show that this algorithm has a runtime with an objective function value that can be equal to or better than DSSA, depending on the adjustment of the dependent parameters. In particular, this method resolves the fairness constraint, which DSSA lacks [20].

(2) The second algorithm is FBIM2, which combines seed selection with the *online processing influence maximization* algorithm, known as OPIMC (see [15]) to ensure maximum coverage and guarantee the fairness constraint. FBIM2 has $O\left(kT \log \frac{n}{\epsilon^2 k}\right)$ complexity. Similar to DSSA, OPIMC does not solve the fairness constraint.

(3) The last is the FBIM3 algorithm, which improves on FBIM2 using the greedy technique with a threshold criterion for selecting a seed set. FBIM3 has $O\left(\frac{T}{\epsilon} \log \frac{k}{\epsilon} \log \frac{n}{\epsilon^2 k}\right)$ complexity. Significantly, the seed set's distribution guarantees a high coverage ratio, which is an expression of ensuring the fairness constraint.

- We further investigate the performance of our algorithms by conducting experiments for the FBIM under both well-known diffusion models, namely, *Linear Threshold* and *Independent Cascade* [3], on real social networks. The results indicate that the seed sets of our algorithms have a coverage ratio with respect to online communities that is greater than the results with OPIMC; specifically, it is 2 to 10 times larger than and there are even cases in which FBIM reaches 100% coverage for the target communities. This process depends on the appropriate parameter selection for each dataset. An extensive coverage ratio signifies the number of chosen seeds covering the target communities, ensuring that the impacted individuals are the ones we want to influence. In addition, the efficiency of FBIM algorithms must be influenced by objective factors due to the implementation method and fairness constraints. This leads to higher cost with a lower objective function value than OPIMC. Nevertheless, the results guarantee the optimal theoretical approximation, especially the fairness constraint.

Organization. The rest of this article includes six sections. Related works is presented in Section 2. Definitions and descriptions related to FBIM are presented in Section 3. Section 4 devises the Threshold Greedy method for the Fair Submodular Maximization problem, that is, the generation of FBIM. Section 5 contains the proposed algorithms and their theoretical analysis. We evaluate the experimental results in Section 6, and conclude the paper in Section 7.

2. Related Works

According to earlier research [3,5,8,21], the IM problem is often addressed using a greedy technique with an approximation of $(1 - 1/e)$. Although the greedy strategy is quite successful for IM, computing the influence function $f(S)$ remains challenging, as it is P-hard. Existing approaches for IM may be classified into three primary classes based on how the influence function is calculated [7].

(1) *Simulation-based approaches*, such as Greedy [3], CELF [22], CELF++ [23,24], and UBLF [25], calculate the influence function using Monte Carlo sampling. To achieve highly scalable algorithms for IM, heuristic techniques can be merged based on a greedy algorithm. These algorithms aim to produce an $(1 - 1/e)$ approximation answer. These methods have an advantage in that they are suitable for diffusion models. However, their disadvantage is that it is necessary to generate many sample cases in order to calculate the objective function with only minor errors. Hence, this approach significantly increases computational costs.

(2) *Proxy-based approaches*, such as SimPath [26], Degree [23], PageRank [27–29], and EasyIM [30], approximate the influence function $f(S)$ to overcome the P-hard problem by devising proxy models. The approximate solution obtained is $(1 - 1/e - \epsilon)$ for any

$\epsilon > 0$. Many algorithms have demonstrated that the proxy-based strategy is efficient for empiricism; however, it does not provide theoretical guarantees.

(3) *Sketch-based approaches*, such as TIM, TIM+ [31], and IMM [32], use a novel RIS sampling method. The goal of these techniques is to produce an $(1 - 1/e - \epsilon)$ -approximate solution with a minimal number of RIS samples [14]. The drawback of these approaches is that their lower bounded budget is unknown, and the number of generated samplings can be rather large. Furthermore, these algorithms guarantee theoretical efficiency thanks to their rigorously bounded solutions and minimal time complexity. However, because they must ensure the approximation ratio for the worst-case scenario, the practical efficiency of sketch-based strategies may be lower than that of proxy-based approaches.

Nguyen et al. [6] have proposed two new sampling techniques, SSA and DSSA, which attempt to obtain a small number of RIS samples while ensuring $(1 - 1/e - \epsilon)$ -approximations. Despite this, Huang et al. [33] discovered that SSA/DSSA has certain flaws. They provided SSA-Fix, an updated version of SSA. Afterwards, Nguyen et al. presented D-SSA-Fix [34], a significantly updated version of DSSA that produces $(1 - 1/e - \epsilon)$ -approximations.

In addition, many studies have sought to resolve other variants of IM [16,31,32,34]. However, almost all of these methods focus on *offline processing*. This means that the user does not receive any output until the final result is obtained. Thus, the user cannot terminate the algorithm early to trade solution quality for efficiency. Motivated by this phenomenon, Tang et al. proposed the OPIMC algorithm for *online processing* of influence maximization in [15]. This method allows the user to terminate any timestamp, then obtain a seed set S and report an approximate guarantee $(1 - 1/e - \epsilon)$, such that S is the IM problem's approximate solution with at least $(1 - \alpha)$ probability, with $\alpha \in (0, 1)$ a user-specified parameter.

Unfortunately, virtually all of the above methods have focused on identifying the most influential nodes to increase the number of nodes affected. This fails to ensure that chosen nodes are evenly distributed across the partitions of the dataset. This shortcoming is a driving force that has attracted the attention of researchers. Recent publications have proposed multiple definitions of fairness and explicitly integrated fairness into the IM problem. One such fairness concern, known as group fairness, is meant to ensure that each group receives a fair share of resources and that the distribution of those resources respects the varied makeup of the groups. These studies of group fairness constraints in the IM problem have obtained positive results.

Tsang et al. (2019) [35] proposed the issue of optimizing the dissemination of a strategy while keeping a group fairness restriction in mind. The authors developed two fairness measures, *maximin fairness* and *group rationality*, to assess group fairness in IM. Maximin fairness measures the smallest number of nodes within each group for whom the influence must be maximized. Meanwhile, the main principle of group rationality is that no group may increase its influence by withdrawing from IM with its proportional allocation of resources and distributing those resources internally. Both measures aim to ensure that each group receives an equitable share of resources.

Stoica et al. (2020) [36] studied the problem of fair resource allocation in influence maximization. The authors provided an algorithmic framework for locating solutions that satisfy fairness constraints for multi-objective submodular maximization problems. This method increases the diversity of nodes in the seed set, and may have an impact on the effectiveness and fairness of the information diffusion process. The authors demonstrated that seeding methods that consider the diversity of nodes in the seed set are more effective and fair in certain circumstances.

Halabi et al. (2020) [37] worked on the problem of maximizing fair submodular functions, including monotone and non-monotone submodular functions, proposing streaming algorithms for this problem. For the monotone case, the authors achieved two results, the $(1/2)$ -approximation algorithm and the $(1/4)$ -approximation algorithm, each of which use $O(\log k)$ time. For non-monotone case, they achieved a $(q + \epsilon)$ -approximation with q

as an input excess ratio, requiring $O(k)$ time. These approaches apply to the creation of fair summaries for massive datasets.

Next, Khajehnejad et al. (2020) [18] studied fair influence maximization in an effort to more fairly reach minority groups. The authors used machine learning approaches to pick a seed set using an adversarial graph embedding technique, which allows for strong impact propagation as well as fairness among communities.

Rahmattalabi et al. (2021) [38] resolved the problem that Tsang et al. studied in [35]. This is the problem of group fairness in the influence maximization problem. However, Rahmattalabi et al. took a different approach, which is to offer a principled characterization of the properties that a fair influence maximization algorithm must meet. The authors designed a framework founded on the social welfare theory that aggregates the cardinal utilities each community derives using isoelastic social welfare functions. In this framework, the trade-off between fairness and efficiency can be handled by a single inequality aversion design parameter.

More recently, Becker et al. (2022) [19] considered the same problem as Tsang et al. [35], proposing a new approach. The authors modeled the problem on the basis of the probabilistic techniques used to choose seed sets, rather than purely deterministic ones. They provided two variations of this probabilistic problem: the node-based problem, which uses probabilistic strategies over nodes, and the set-based problem, which uses probabilistic strategies over sets of nodes. After examining the relationship between the two probabilistic problems, the authors demonstrated that both probabilistic variants provide approximation algorithms that achieve a constant multiplicative factor of $(1 - 1/e)$, minus an arbitrarily small additive error caused by the simulation of the information spread.

Ali et al. (2022) [39] solved the fairness of the spread process in various groups. They focused on the time-critical aspect of IM, examining the number of affected people and the time step at which they are influenced. Subsequently, they found that maximizing the expected number of individuals reached by selecting a fixed cardinality seed set and minimizing the amount of seeds needed to affect a given portion of the network can lead to unfair solutions. The authors proposed an objective function that balances two goals, such as maximizing the expected number of nodes reached and minimizing the maximum disparity in influence between any two communities.

Razaghi et al. (2022) [40] worked on the same group fairness-aware influence maximization problem in social networks as Tsang et al. [35]. However, they fixed an important omission that Tsang et al. overlooked by assessing the time required to receive resources by different nodes of groups. The authors expanded the concept of group fairness in the IM problem by examining the speed of node activation in different social network groups. They proposed a multi-objective meta-heuristic (SetMOGWO) founded on the multi-objective gray wolf optimizer in order to increase the fair propagation of information in the IM problem as it relates to various fairness measures.

Based on the available literature, the above publications show that previous studies have been concerned with particular aspects or constraints on the problem of group fairness in IM. As far as we know, there has been no single study that mentions the fairness constraints on both the lower and upper bounded budgets of the target communities in the dataset, which is the reason and impetus for our to research and our attempts find a solution to this shortcoming.

3. Preliminaries

In this section, we first introduce the definition of the Fair Submodular Maximization (FSM) problem, which was studied by Halabi et al. [37], generate the present problem, and recap the Fair Greedy algorithm, which is able to return an approximation of $(1/2)$ within $O(nk)$ queries. We then introduce our studied fairness influence maximization problem and recap certain properties of the problem. Table 1 summarizes the notations commonly used in this paper.

Table 1. Table of the usual notation used in the maximizing submodular problem under fairness constraint.

Notation	Description
n	the number of nodes in the graph.
V	the set of nodes in the graph G , $ V = n$.
2^V	the subset family of V .
m	the number of edges in the graph.
E	the set of edges in the graph G , $ E = m$.
w	the set of edge weights in the graph.
v	a random node in the graph.
u	a neighbor node of v in the graph.
k	a total budget, which is upper bound of $ S $.
K	the number of target communities is selected for the FBIM's input.
\mathcal{C}'	the set of K communities in network G .
\mathbf{C}	a set of disjoint communities of the graph, $ \mathbf{C} = N$
N	the size of a set \mathbf{C}
C_i, C_j	the i -th community and the j -th community.
V_i	the set of nodes of the community C_i
k_i^l	the lower bounded budget of the community C_i .
k_i^u	the upper bounded budget of the community C_i .
S	the returned size- k seed set of algorithms.
S^*	an optimal size- k seed set.
R_i, R_j	a random RR set.
T	the number of nodes in \mathcal{C}' , $T = \left \bigcup_{C_i \in \mathcal{C}'} (C_i) \right $ and $T \leq n$.
$\mathcal{R}, \mathcal{R}', \mathcal{R}_1, \mathcal{R}_2$	the set of random RR sets.
$\text{Cov}_{\mathcal{R}}(S)$	number of RR sets in \mathcal{R} incident at some node in S .
$f(S)$	influence spread of a seed set S
$f_l(S), f_u(S^*)$	the lower bound of $f(S)$ and the upper bound of $f(S^*)$.
$\hat{f}(S)$	an estimation of $f(S)$ on a collection of RR sets \mathcal{R}
\mathbb{E}	an expected value

Consider a ground set V of n elements and a submodular function $f : 2^V \rightarrow \mathbb{R}_+$; given two sets $S, T \subseteq V$, the marginal gain of S in relation to T is defined as follows:

$$f(S|T) = f(S \cup T) - f(T) \tag{2}$$

For two sets $S, T, S \subseteq T \subseteq V$ and an element $v \in V \setminus T$, the function f is *submodular* if

$$f(v|S) \geq f(v|T), \tag{3}$$

while f is *monotone* if $f(T) \geq f(S)$. The FSM problem is defined as follows:

Definition 1 (FSM problem). Assume that V is divided into \mathcal{C}' disjoint subsets V_1, V_2, \dots, V_K , and $C_i \cap C_j = \emptyset$. Each subset V_i has a lower and an upper bound on the number of elements k_i^l and

k_i^u that represents the fairness constraint. Let a positive number k be a global cardinality constraint. The FSM then seeks to find

$$\max: f(S) \tag{4}$$

$$\text{subject to: } |S| \leq k \tag{5}$$

$$k_i^l \leq |S \cap V_i| \leq k_i^u, \quad i = 1 \dots K \tag{6}$$

We define an instance of the problem FSM as a tuple $(f, V, V_1, \dots, V_K, k)$. The FSM problem can then be reduced to a submodular maximization under a matroid constraint (SMM) problem [37], defined as follows:

Definition 2 (SMM problem). *The problem seeks to select a set $S \subseteq V$ with $S \in \mathcal{M}$ such that it maximizes $f(S)$, where \mathcal{M} is a matroid. A family of sets \mathcal{M} with subset 2^V is referred to as a matroid if it satisfies the following conditions:*

1. $\mathcal{M} \neq \emptyset$;
2. downward-closedness, i.e., if $A \subseteq B$ and $B \in \mathcal{M}$, then $A \in \mathcal{M}$;
3. augmentation, i.e., if $A, B \in \mathcal{M}$ with $|A| \leq |B|$, then there exists $s \in B$ such that $A + s \in \mathcal{M}$.

Fair Greedy Algorithm. We now recap the Fair Greedy algorithm (Algorithm 1) [37]. The operation of this algorithm is based on the notation of an extendable set.

Algorithm 1: Fair Greedy Algorithm

Input: An instance of FSM problem $(f, V, V_1, \dots, V_K, k)$

Output: A 1/2-approximation solution S

1. $S \leftarrow \emptyset$
 2. **for** $i = 1$ **to** k **do**
 3. $U \leftarrow \{s \in V \mid S \cup s \text{ is extendable} \}$
 4. $v \leftarrow \arg \max_{s \in U} f(s \mid S)$
 5. $S \leftarrow S \cup \{v\}$
 6. **return** S
-

Definition 3 (An extendable set [37]). *A set $S, S \subseteq V$ is extendable if and only if*

$$|S \cap C_i| \leq k_i^u, \quad i = 1, \dots, K \tag{7}$$

and

$$\sum_{i=1}^K \max(|S \cap C_i|, k_i^l) \leq k \tag{8}$$

Fairness Budget Influence Maximization Problem.

In this section, we introduce the FBIM problem, which is the focus of this study, and its related definitions. FBIM inherits two issues: *fair submodular maximization* (FSM) [37] and *fair influence maximization* (FIM) [38].

Definition 4 (FIM problem). *Consider a graph $G = (V, E)$, where V is the set of vertices, $|V| = n$, E is the set of edges, $|E| = m$, and $k \in \mathbb{Z}^+$ is a global cardinality constraint. Let \mathbf{C} be a set of disjoint communities (empty intersections) of the graph. Each vertex v of V belongs to one of the communities $C_i \in \mathbf{C}$, $i \in 1, \dots, N$ such that $V_1 \cup \dots \cup V_N = V$, where V_i denotes the set of vertices of the community C_i . Furthermore, communities may be disconnected, that is, $\forall C_i, C_j \in \mathbf{C}$ and $\forall v \in V_i, u \in V_j$; in this case, there is no edge between v and u . Here, \mathcal{A} denotes the initial set of vertices (referred to as influencer vertices), and we define $\mathcal{A}^* := \{A \subset V \mid |A| \leq k\}$ as the set of*

feasible budget influencers. Lastly, for any choice of $\mathcal{A} \in \mathcal{A}^*$, we let $h_{C_i}(\mathcal{A})$ denote the expected fraction of the influenced vertices of a community C_i , where the expectation in terms of the spread of influence is reached under a diffusion model. The fair influence maximization problem solves the optimization problem

$$\underset{\mathcal{A} \in \mathcal{A}^*}{\text{maximize}} \sum_{C_i \in \mathcal{C}} |V_i| \cdot h_{C_i}(\mathcal{A}) \tag{9}$$

Definition 5 (FBIM problem). Consider a social network $G = (V, E, w)$, where $|V| = n$, and a set $\mathcal{C}' = \{C_1, C_2, \dots, C_K\}$, where $\forall C_i \subset G$ and $C_i \cap_{i \neq j} C_j = \emptyset$. Each community C_i has a pair of upper k_i^u and lower k_i^l bounded budgets where $k_i^l \leq k_i^u$. For a given total budget k , the problem seeks to find a seed set S , $|S| \leq k$, which satisfies $k_i^l \leq |S \cap C_i| \leq k_i^u$ such that $f(S)$ is maximal, where $f(S)$ is the influence function. Here, $f(S)$ measures the expected number of users in V that can be influenced by the elements under S in a diffusion model.

In this paper, we solve the FBIM problem under both the *Linear Threshold* (LT) and *Independent Cascade* (IC) models, which are the well-known and standard diffusion models for the IM problem as well as other problems related to the analysis of social networks [6,15,41]. These models were first proposed by Kempe et al. in [3].

Definition 6 (LT model). Consider a directed graph $G = (V, E, w)$, where V is the set of vertices, $|V| = n$, and E is the set of edges, $|E| = m$. A vertex v in G is affected by each of its neighbors u with probability $p(u, v)$, $p(u, v) \in w$, and $\sum_{u \text{ neighbor of } v} p(u, v) \leq 1$. Every vertex v in V is assigned a threshold Λ_v from the interval $[0; 1]$ at random. This threshold is the weighted proportion of v 's neighbors (active vertices) influencing v to become active. The process spreads as follows. Originally, we initialize a random set of threshold values and a seed set of active vertices S while the rest are inactive. In the t -th step, all active vertices in the $(t - 1)$ -th step remain active, and the process activates any vertex v with a total weight of active neighbors greater than or equal to Λ_v :

$$\sum_{u \text{ active neighbor of } v} p(u, v) \leq 1 \tag{10}$$

Definition 7 (IC model). Consider a directed graph $G = (V, E, w)$, where V is the set of vertices, $|V| = n$, and E is the set of edges $|E| = m$. At first, the vertices in the seed set S are active, while all remaining nodes are inactive. The process spreads according to the following rule. At the t -th step, if vertex v becomes active first, it has only one chance to activate each neighbor u . The probability of success is $p(v, u)$, $p(v, u) \in w$. If u has multiple active neighbors, their diffusion is sequenced in random order. As soon as vertex v succeeds, vertex u becomes active in the $(t + 1)$ -th step. Regardless of whether v succeeds or not, it cannot make any further attempts to activate u in subsequent rounds. In this way, the process works until no more activation is feasible.

For the FBIM problem, according to the proposition of Halabi et al. in [37], the greedy method selects the element with the highest *marginal gain*, where the marginal gain of an element s is the value of $(f(S \cup s) - f(S))$, while meeting the constraints. We note that the greedy approach might not produce a feasible solution if this element was only required to hold the cardinality and upper bounds constraints, as it may meet the global cardinality restriction without meeting the lower bounds. As a result, more careful selection of elements is required. Thus, the seed set S must be an extendable set.

In addition, there are several definitions that are typically used in this paper. These are *Reverse Influence Sampling* (RIS), *Reachable Reverse Sets* (RR set), and the *Coverage of seed set S on the set of Reachable Reverse Sets* ($\text{Cov}_{\mathcal{R}}(S)$).

Definition 8 (RIS [14]). Given a graph $G = (V, E, w)$, RIS apprehends the influence scene of G by generating a set \mathcal{R} of random RR sets. An RR set contains nodes that can reach v in g , where v is a random node in V and g is a sample graph from G .

Definition 9 (RR set [14]). Consider a graph $G = (V, E, w)$ under the IC model; a random RR set R_i is generated from G according to the following steps:

- Step 1. Choose a source vertex $v \in V$;
- Step 2. Generate a sample graph g from G ;
- Step 3. Return R_i such that it contains vertices that can be reached from v in g .

We refer to the set of random RR sets as \mathcal{R} . According to [6], finding a seed set S and influence spread $f(S)$ is based on computing the coverage of S on the majority of RR sets. Because of the generation of a set \mathcal{R} of multiple random RR sets, influential vertices may commonly occur in the RR sets. Hence, we can find the vertices that occur in the majority of RR sets in order to add the seed set S . Moreover, the influence spread $f(S)$ on a random RR set R_i is proportional to the probability that S intersects with R_i . Therefore, a seed set S covers a set RR R_i if $S \cap R_i \neq \emptyset$. For greater simplicity, we refer to the coverage of S on \mathcal{R} as $\text{Cov}_{\mathcal{R}}(S)$, and compute it as follows:

$$\text{Cov}_{\mathcal{R}}(S) = \sum_{R_i \in \mathcal{R}} \min\{1, |S \cap R_i|\} \tag{11}$$

Furthermore, the influence spread $f(S)$ on a random RR set R_i is proportional to the probability of S intersecting with R_i . Borgs et al. [14] proposed the following calculation for the influence spread function $f(\cdot)$:

$$f(S) = n \cdot \mathbb{E}[\min\{1, |S \cap R_i|\}] \tag{12}$$

and for the estimation of $f(S)$ over a collection of RR sets \mathcal{R} ,

$$\hat{f}(S) = n \cdot \text{Cov}_{\mathcal{R}}(S) / |\mathcal{R}| \tag{13}$$

4. Threshold Greedy for FSM

In this section, we devise the Threshold Greedy algorithm, which applies the decreasing threshold greedy strategy for the FSM problem. This algorithm improves on Algorithm 1 by using a decreasing threshold to reduce the number of data traversals while continuing to ensure the approximate solution.

4.1. Algorithm Description

The Threshold Greedy algorithm operates as follows. At the beginning, it initiates the initial threshold t with M , where $M = \max_{u \in V} f(u)$. One loop of the inner loop ‘for’ is named one *iteration*. It scans all elements in S . At each iteration, if the current element s satisfies two conditions, $S \cup s$ is extendable and $f(s|S) \geq t$, then s is selected into S . After each loop of the “while” loop, t is decremented by $(1 - \epsilon)$ times until $t < \epsilon M/k$, then the algorithm terminates and returns S . A detailed description of this algorithm is provided in Algorithm 2.

4.2. Theoretical Analysis

The following theoretical analysis along with the proofs in Lemma 1 and Theorem 1 demonstrate the feasibility and efficiency of this algorithm in guaranteeing the approximation solution.

Algorithm 2: Threshold Greedy Algorithm

Input: An instance of FSM problem $(f, V, V_1, \dots, V_K, k)$, accuracy parameter ϵ .

Output: A $(1/2 - \epsilon)$ -approximation solution S

1. $S \leftarrow \emptyset$
 2. $M \leftarrow \max_{u \in V} f(u)$
 3. $t \leftarrow M$
 4. **while** $t \geq \epsilon M/k$ **do**
 5. **for** $s \in V$ **do**
 6. **if** $S \cup \{s\}$ is extendable and $f(s|S) \geq t$ **then**
 7. $S \leftarrow S \cup \{s\}$
 8. $t \leftarrow (1 - \epsilon)t$
 9. **return** S
-

Lemma 1. Denote t_i as threshold t at the i -th iteration and let S_i be S at the beginning of the i -th iteration. We first show that

$$t_i \geq (1 - \epsilon) \max_{s \in V: S_i \cup \{s\} \text{ is extendable}} f(s|S_i) \tag{14}$$

Proof of Lemma 1. We prove Lemma 1 using induction. If $i = 1$, $S_1 = \emptyset$ and $t_1 = M \geq (1 - \epsilon) \max_{s \in V: S_1 \cup \{s\} \text{ is extendable}} f(s)$. Assume that the lemma holds for $i \geq 1$. Then, we have

$$t_{i+1} = (1 - \epsilon)t_i \geq (1 - \epsilon) \max_{s \in V: S_{i+1} \cup \{s\} \text{ is extendable}} f(s|S_{i+1}) \tag{15}$$

The inequality is due to the fact that the element

$$s_{max} = \arg \max_{s \in V: S_{i+1} \cup \{s\} \text{ is extendable}} f(s|S_{i+1})$$

was not added to S in iteration i . The lemma is proved. \square

Theorem 1. Algorithm 2 requires $O(\frac{n}{\epsilon} \log \frac{k}{\epsilon})$ runtime and returns a $(1/2 - \epsilon)$ -approximation solution for the FSM problem.

Proof of Theorem 1. The computational complexity can be easily proven. Assume that there are a total of x number of iterations in the “while” loop of Algorithm 2. Therefore, we have $(1 - \epsilon)^x = \frac{\epsilon}{k}$. Solving this equation yields

$$x = \frac{\log \frac{k}{\epsilon}}{\log \frac{1}{1-\epsilon}} \leq \frac{1}{\epsilon} \log \frac{k}{\epsilon} \tag{16}$$

The “for” loop carries out n iterations. Thus, the time complexity of this algorithm is $O(\frac{n}{\epsilon} \log \frac{k}{\epsilon})$.

In addition, assuming that $s_{max} = \max_{s \in V: S_i \cup \{s\} \text{ is extendable}} f(s|S_i)$, $S_i = S_{i-1} \cup \{s_1, s_2, \dots, s_l\}$, that S^s is S immediately before s is processed, and that $S_i^s \cup \{s\}$ is extendable, we have the following:

$$t_i \geq f(s_{max}|S^s) \geq f(s_{max}|S_i) \tag{17}$$

Now, assume that $S = \{s_1, s_2, \dots, s_u\}, u \leq k$ is S after ending the main loop of the algorithm. We can consider the following two cases:

Case 1. With $|S| = u, u = k$, assume that $O = \{o_1, o_2, \dots, o_k\}$ is the optimal solution such that $\{s_1, s_2, \dots, s_{i-1}, o_i\}$ can be extended, which exists because the extendability is matroid [37] and satisfies the augmentation property. Denote S^i to be S immediately before s_i is added to S , and let $t(s_i)$ be t at iteration s_i to be added to S . We then have the following:

$$\begin{aligned}
 f(S) &= \sum_{i=1}^u f(s_i | \{s_1, s_2, \dots, s_{i-1}\}) \\
 &\geq \sum_{i=1}^u t(s_i) \\
 &\geq \sum_{i=1}^u (1 - \epsilon) f(s_i | S^i) \text{ (By Lemma 1)} \\
 &\geq \sum_{i=1}^u (1 - \epsilon) f(o_i | S^i) \\
 &\geq \sum_{i=1}^u (1 - \epsilon) f(o_i | \{s_1, s_2, \dots, s_{i-1}\}) \\
 &\geq \sum_{i=1}^u (1 - \epsilon) f(o_i | \{s_1, s_2, \dots, s_u\} \cup \{o_1, \dots, o_{i-1}\}) \text{ (Due to the submodularity of } f) \\
 &= \sum_{i=1}^u (1 - \epsilon) f(o_i | S \cup \{o_1, \dots, o_{i-1}\}) \\
 &= (1 - \epsilon)(f(O \cup S) - f(S)) \text{ (by equality (2))}
 \end{aligned}$$

and thus,

$$\begin{aligned}
 f(S) &\geq \frac{(1 - \epsilon)}{(2 - \epsilon)} f(O) \\
 &\geq \frac{(1 - 2\epsilon)}{2} f(O) \\
 &= \left(\frac{1}{2} - \epsilon\right) f(O)
 \end{aligned}$$

Case 2. With $|S| = u, u < k$ and $S = \{s_1, s_2, \dots, s_u\}$, assume that O is the optimal solution, $S' = \{s_{u+1}, \dots, s_k\}$, and $S_k = \{s_1, s_2, \dots, s_u\} \cup \{s_{u+1}, \dots, s_k\}$, meaning that $S_k = S \cup S'$. Denote t_{last} to be t after ending the main loop of the algorithm. Then, we have the following:

$$\begin{aligned}
 \sum_{i=u+1}^k f(s_i | \{s_1, s_2, \dots, s_{i-1}\}) &\leq kt_{last} \text{ (Due to } f(s_i | S)_{i=u+1, \dots, k} \leq t_{last}) \\
 &\leq k \frac{\epsilon M}{k} \leq \epsilon M \leq \epsilon f(O)
 \end{aligned}$$

and thus,

$$f(S_k) - f(S) \leq \epsilon f(O)$$

We now have the equivalent inequality, as follows:

$$\begin{aligned}
 f(S) &\geq f(S_k) - \epsilon f(O) \\
 &\geq \left(\frac{1}{2} - \epsilon\right) f(O) - \epsilon f(O) \text{ (By the proof of Case 1)} \\
 &\geq \left(\frac{1}{2} - 2\epsilon\right) f(O) \\
 &\geq \left(\frac{1}{2} - \epsilon'\right) f(O) \text{ (with } \epsilon' = 2\epsilon)
 \end{aligned}$$

The proof is completed. \square

As mentioned above, FSM is the generation of the FBIM problem. Therefore, the Threshold Greedy algorithm is the basis of an important key to designing an efficient algorithm for the FBIM problem in the next section. This is the Threshold Greedy Algorithm for Fairness Max Cover, which finds a seed set S such that it maximizes the coverage and satisfies the fairness constraint.

5. Proposed Algorithms for FBIM

This section introduces three algorithms for the FBIM problem, called FBIM1, FBIM2, and FBIM3. The details of these algorithms are fully presented in Algorithms 3–7. Because our proposed algorithms are based on the DSSA [6] and OPIMC [15] methods, we do not perform the proofs provided in the originals.

5.1. FBIM1—An Algorithm Based on the Stop-and-Stare Method

This method combines an improved greedy strategy for selecting seeds satisfying the fairness constraint with generating sampling of the DSSA. The FBIM1 algorithm’s fundamental principle is to (1) choose k nodes that occur on the majority of communities and add them to S to maximize the coverage of S on the set of RR sets; and (2) compute $f(S)$ using the stop-and-stare technique. If the result does not satisfy the threshold condition of the algorithm, the process repeats the search for a new S on the new set of RR sets with double the number of elements. The details of this algorithm are fully presented in Algorithms 3 and 4.

Algorithm 3: Fairness-Max-Coverage (FMC) procedure

Input: Graph G , \mathcal{C}' , k , RR sets (\mathcal{R}).

Output: A seed set S that satisfies fairness constraint, $|S| \leq k$ and its estimated influence $f(S)$.

1. $S \leftarrow \emptyset$
 2. **for** $i = 1$ **to** k **do**
 3. $\mathbb{U} \leftarrow \{s \in (C_j_{C_j \in \mathcal{C}'} \setminus S) \mid S \cup s \text{ is extendable} \}$
 4. $v \leftarrow \arg \max_{\{s \in \mathbb{U}\}} (\text{Cov}_{\mathcal{R}}(S \cup s) - \text{Cov}_{\mathcal{R}}(S))$
 5. $S \leftarrow v$
 6. **return** S .
-

5.1.1. Algorithm Description

The FBIM1 algorithm inputs a K -size communities set \mathcal{C}' , a budget k , parameters ϵ and δ , ($0 \leq \epsilon, \delta \leq 1$), which are related to the solution’s quality, and the algorithm’s runtime. This allows us to adjust the solution quality trade-off with runtime. In particular, ϵ and δ guarantee the size bound of S . In contrast, k and the set \mathcal{C}' guarantee the value of the objective function $f(S)$ and the coverage ratio of S in communities in \mathcal{C}' , which is considered to satisfy the fairness constraint.

In the initial step, the algorithm generates a set \mathcal{R} that contains Γ random RR sets by RIS, with the value of Γ initialized as line 1. The formula of Γ has been proven in [6]. Subsequently, based on this \mathcal{R} , Algorithm 3 returns a seed set S .

Algorithm 3 receives a K -size community \mathcal{C}' set, a budget k and a set \mathcal{R} . This algorithm produces a k -size seed set S that satisfies fairness constraint. Initiate an empty set S . It performs a loop of k times, each iteration finds an element s in all C_j of \mathcal{C}' , except in S , so that $(S + s)$ is extendable and the coverage of s on the RR sets of \mathcal{R} is maximal. This element s is added to S . After the loop ends, the algorithm returns the seed set S .

Algorithm 4: FBIM1 algorithm

Input: Graph G , \mathcal{C}' , k , and $0 \leq \epsilon, \delta \leq 1$
Output: A seed set S , $|S| \leq k$.

1. $\Gamma \leftarrow \sqrt{8}(1 + \epsilon)^2 \ln \frac{2}{\delta} \frac{1}{\epsilon^2}$
2. $\mathcal{R} \leftarrow$ Generate Γ random RR sets by RIS
3. $S \leftarrow \text{FMC}(G, \mathcal{C}', k, \mathcal{R})$
4. $f(S) \leftarrow n \cdot \text{Cov}_{\mathcal{R}}(S) / |\mathcal{R}|$
5. **repeat**
6. $\mathcal{R}' \leftarrow$ Generate $|\mathcal{R}|$ random RR sets by RIS
7. $f'(S) \leftarrow n \cdot \text{Cov}_{\mathcal{R}'}(S) / |\mathcal{R}'|$
8. $\epsilon_1 \leftarrow (f(S) / f'(S)) - 1$
9. **if** $\epsilon_1 \leq \epsilon$ **then**
10. $\epsilon_2 \leftarrow (\epsilon - \epsilon_1) / (2(1 + \epsilon_1))$
11. $\epsilon_3 \leftarrow (\epsilon - \epsilon_1) / (2(1 - \frac{1}{\epsilon}))$
12. $\delta_1 \leftarrow e \frac{\text{Cov}_{\mathcal{R}}(S) \cdot \epsilon_3^2}{\sqrt{8}(1 + \epsilon_1)(1 + \epsilon_2)}$
13. $\delta_2 \leftarrow e \frac{(\text{Cov}_{\mathcal{R}'}(S) - 1) \cdot \epsilon_2^2}{\sqrt{8}(1 + \epsilon_2)}$
14. **if** $\delta_1 + \delta_2 \leq \delta$ **then**
15. **return** S .
16. $\mathcal{R} \leftarrow \mathcal{R} \cup \mathcal{R}'$
17. $S \leftarrow \text{FMC}(G, \mathcal{C}', k, \mathcal{R})$
18. $f(S) \leftarrow n \cdot \text{Cov}_{\mathcal{R}}(S) / |\mathcal{R}|$
19. **until** $|\mathcal{R}| \geq (8 + 2\epsilon)n \frac{\ln \frac{2}{\delta} + \ln \binom{n}{k}}{\epsilon^2}$;
20. **return** S .

We now turn our attention to Algorithm 4. This algorithm executes an indefinite loop; in each iteration, it evaluates the efficiency of S by computing $f'(S)$ on set \mathcal{R}' and checks whether $(f(S) / f'(S)) - 1$ meets the conditions in lines 9 and 14. If this is true, the algorithm stops. Otherwise, \mathcal{R} doubles by adding \mathcal{R}' to find a new S and proceeds to the next iteration. The loop stops when $|\mathcal{R}|$ is at least $(8 + 2\epsilon)n \cdot \frac{\ln \frac{2}{\delta} + \ln \binom{n}{k}}{\epsilon^2}$.

5.1.2. Theoretical Analysis

Theorem 2. Algorithm 3 is an improved greedy algorithm; it has $O(kT)$ complexity, with $T = \left| \bigcup_{C_i \in \mathcal{C}'} (C_i) \right|$ and $T \leq n$.

Proof of Theorem 2. Algorithm 3 iterates k times to select seeds for the seed set S of size k . Each iteration scans the majority of the elements of C_i ($\forall C_i \in \mathcal{C}'$) to select an element that has the greatest coverage on \mathcal{R} . As a result, the complexity of the algorithm is $O(kT)$. □

Theorem 3. The complexity of Algorithm 4 is $O\left(kT \log\left((8 + 2\epsilon)n \frac{\ln \frac{2}{\delta} + \ln \binom{n}{k}}{\epsilon^2}\right)\right)$.

Proof of Theorem 3. Algorithm 4 iterates the generation of the set \mathcal{R} of random RR sets with randomly chosen source elements from C_i and finds $\langle S, f(S) \rangle$ on the basis of \mathcal{R} using Algorithm 3. At lines 14 and 19, Algorithm 4 has two requirements to break the loop. In each iteration, $|\mathcal{R}|$ is doubled in size and calls Algorithm 3 to find a new S ; thus, in the worst case,

this algorithm stops when it meets condition line 19, which indicates that the maximum number of iterations is $\log\left((8 + 2\epsilon)n \frac{\ln \frac{2}{\delta} + \ln \binom{n}{k}}{\epsilon^2}\right)$. Consequently, the complexity of this method is $O\left(kT \log\left((8 + 2\epsilon)n \frac{\ln \frac{2}{\delta} + \ln \binom{n}{k}}{\epsilon^2}\right)\right)$. \square

5.2. FBIM2 & FBIM3—Algorithms Based on the Online Processing of Influence Maximization Method

5.2.1. Algorithm Description

The FBIM2 and FBIM3 algorithms are exactly similar in their main idea, and are only different in the way they find the seed set S . The main idea of these algorithms is to perform a finite loop that performs the following processing: (1) generate two set of RR sets, i.e., \mathcal{R}_1 and \mathcal{R}_2 ; (2) find a seed set S based on \mathcal{R}_1 such that S satisfies the fairness constraint; (3) compute $f_l(S)$ based on \mathcal{R}_2 and $f_u(S^*)$ based on \mathcal{R}_1 . If the ratio $f_l(S)/f_u(S^*)$ satisfies the optimal solution approximation (e.g., for FBIM2, it is at least $1/2 - \epsilon$, and for FBIM3, it is at least $1/2 - 2\epsilon$), the algorithm stops before the limit. Otherwise, the algorithm repeats the step in which it finds S , now with \mathcal{R}_1 and \mathcal{R}_2 doubled in size.

As mentioned above, FBIM2 and FBIM3 are inherited from the OPIMC method of Tang et al. [15]; thus, we have the expressions $f_l(S)$ and $f_u(S')$ as follows:

$$f_l(S) = \left(\left(\sqrt{\text{Cov}_{\mathcal{R}_2}(S) + \frac{2\ln(1/\delta_2)}{9}} - \sqrt{\frac{\ln(1/\delta_2)}{2}} \right)^2 - \frac{\ln(1/\delta_2)}{18} \right) \cdot \frac{n}{\Gamma_0} \tag{18}$$

$$f_u(S^*) = \left(\sqrt{\frac{\text{Cov}_{\mathcal{R}_1}(S)}{1/2} + \frac{\ln(1/\delta_1)}{2}} + \sqrt{\frac{\ln(1/\delta_1)}{2}} \right)^2 \cdot \frac{n}{\Gamma_0} \tag{19}$$

For FBIM2, finding the seed set S is similar to FBIM1 (Algorithm 3), that is, k elements are selected in all C_i communities of \mathcal{C}' , meaning that each element appears the most in communities and guarantees an extendable S after adding to it, which is known as the mandatory condition of the fairness constraint. The details of this algorithm are fully presented in Algorithm 5.

For FBIM3, finding the seed set S is improved by decreasing the search elements times for S . Instead of having to perform k iterations, as FBIM2 does, FBIM3 only requires at most $\frac{1}{\epsilon} \log\left(\frac{k}{\epsilon}\right)$ iterations. The detail of this process is fully presented in the Threshold Greedy Algorithm for Fairness Max Cover procedure (Algorithm 6). The main idea of this algorithm is to find elements s in C_j , ($\forall C_j \in \mathcal{C}'$) that make $S + s$ is extendable while maximizing the gain of coverage ratio of s when adding to S on \mathcal{R} (line 4 and 5). This idea is based on the principle of a simple near-linear time algorithm for the problem of maximizing monotone submodular functions, which was studied by Badanidiyuru and Vondrák [42].

Algorithm 5: FBIM2 algorithm

Input: Graph $G, \mathcal{C}', k, \epsilon, \delta$

Output: A solution $S, |S| \leq k$

- $$2n \left(\frac{1}{2} \sqrt{\ln \frac{6}{\delta}} + \sqrt{\frac{1}{2} (\ln \binom{n}{k} + \ln \frac{6}{\delta})} \right)^2$$
1. $\Gamma_{max} \leftarrow \frac{2n \left(\frac{1}{2} \sqrt{\ln \frac{6}{\delta}} + \sqrt{\frac{1}{2} (\ln \binom{n}{k} + \ln \frac{6}{\delta})} \right)^2}{\epsilon^2 k}$
 2. $\Gamma_0 \leftarrow \Gamma_{max} \cdot \epsilon^2 k / n$
 3. $\mathcal{R}_1 \leftarrow$ Generate Γ_0 random RR sets by RIS
 4. $\mathcal{R}_2 \leftarrow$ Generate Γ_0 random RR sets by RIS
 5. $i_{max} \leftarrow \lceil \log_2(\Gamma_{max} / \Gamma_0) \rceil$
 6. **for** $i \leftarrow 1$ **to** i_{max} **do**
 7. $S \leftarrow$ FMC($G, \mathcal{C}', k, \mathcal{R}_1$)
 8. $\delta_1 = \delta_2 = \delta / (3i_{max})$
 9. compute $f_l(S)$ and $f_u(S^*)$ by (18) and (19)
 10. $\epsilon' \leftarrow f_l(S) / f_u(S^*)$
 11. **if** $\epsilon' \geq (1/2 - \epsilon)$ **or** $i == i_{max}$ **then**
 12. **return** S .
 13. $\Gamma_0 \leftarrow 2\Gamma_0$ and generate Γ_0 new random RR sets for $\mathcal{R}_1, \mathcal{R}_2$.
-

Algorithm 6: Threshold Greedy Algorithm for Fairness Max Cover (ThresholdGreedy(\mathcal{R}, ϵ)) procedure

Input: Graph $G, \mathcal{C}', k, \mathcal{R}$ sets (\mathcal{R}), $\epsilon > 0$

Output: An $(1/2 - \epsilon)$ -optimal solution, $S, |S| \leq k$

1. $S \leftarrow \emptyset$
 2. $M \leftarrow \max_{u \in V} \text{Cov}_{\mathcal{R}}(u)$
 3. **for** ($t = M; t \geq \epsilon M / k; t \leftarrow (1 - \epsilon)t$) **do**
 4. **for** $s \in (C_{j \in \mathcal{C}'} \setminus S) | S + s$ is extendable **do**
 5. **if** $(\text{Cov}_{\mathcal{R}}(S \cup s) - \text{Cov}_{\mathcal{R}}(S)) \geq t$ **then**
 6. $S \leftarrow S \cup \{s\}$
 7. **return** S .
-

5.2.2. Theoretical Analysis

Theorem 4. Algorithm 5 has $O\left(kT \log \frac{n}{\epsilon^2 k}\right)$ complexity.

Proof of Theorem 4. When Algorithm 5 iterates, it generates two random RR sets, \mathcal{R}_1 and \mathcal{R}_2 , with each set having a size of Γ_0 . Later, the algorithm finds a seed set S based on \mathcal{R}_1 through Algorithm 3. Next, it calculates the approximation solution ϵ' of S , with $\epsilon' = \frac{f_l(S)}{f_u(S^*)}$. This algorithm has two conditions to stop and return the result on line 11.

In each iteration, Γ_0 doubles in size and calls Algorithm 3 to find a new S ; thus, in the worst case, this algorithm stops when it has not found S satisfying $\epsilon' \geq (1/2 - \epsilon)$ and the number of iterations has reached i_{max} , with $i_{max} = \lceil \log(\Gamma_{max} / \Gamma_0) \rceil$ in line 5. In such a case, we have

$$\Gamma_{max} = \frac{2n \left(\frac{1}{2} \sqrt{\ln \frac{6}{\delta}} + \sqrt{\frac{1}{2} (\ln \binom{n}{k} + \ln \frac{6}{\delta})} \right)^2}{\epsilon^2 k} \text{ (at line 1) and } \Gamma_0 = \Gamma_{max} \cdot \epsilon^2 k / n \text{ (at line 2).}$$

After calculating to reduce for i_{max} , we have $i_{max} = \lceil \log \frac{n}{\epsilon^2 k} \rceil$. As a result, this algorithm has $O\left(kT \log \frac{n}{\epsilon^2 k}\right)$ complexity. \square

Theorem 5. Algorithm 6 has $O\left(\frac{T}{\epsilon} \log \frac{k}{\epsilon}\right)$ complexity.

Proof of Theorem 5. Similar to Algorithm 2, in Algorithm 6 the outer “for” loop requires at most $\frac{1}{\epsilon} \log \frac{k}{\epsilon}$ iterations, while the inner “for” loop requires a maximum of T iterations. As a result, this algorithm has $O\left(\frac{T}{\epsilon} \log \frac{k}{\epsilon}\right)$ complexity. \square

Theorem 6. Algorithm 7 has $O\left(\frac{T}{\epsilon} \log \frac{k}{\epsilon} \log \frac{n}{\epsilon^2 k}\right)$ complexity.

Proof of Theorem 6. Algorithm 7 operates similarly to Algorithm 5, only differing in the step in which it finds set S (in line 7); that is, Algorithm 7 uses Algorithm 6. Therefore, we can likewise prove that for Algorithm 5, Algorithm 7 has $O\left(\frac{T}{\epsilon} \log \frac{k}{\epsilon} \log \frac{n}{\epsilon^2 k}\right)$ complexity. \square

Algorithm 7: FBIM3 algorithm

Input: Graph $G, \mathcal{C}', k, \epsilon, \delta$

Output: A solution $S, |S| \leq k$

1. $\Gamma_{max} \leftarrow \frac{2n \left(\frac{1}{2} \sqrt{\ln \frac{6}{\delta}} + \sqrt{\frac{1}{2} (\ln \binom{n}{k} + \ln \frac{6}{\delta})} \right)^2}{\epsilon^2 k}$
 2. $\Gamma_0 \leftarrow \Gamma_{max} \frac{\epsilon^2 k}{n}$
 3. $\mathcal{R}_1 \leftarrow$ Generate Γ_0 random RR sets by RIS
 4. $\mathcal{R}_2 \leftarrow$ Generate Γ_0 random RR sets by RIS
 5. $i_{max} \leftarrow \lceil \log_2(\Gamma_{max}/\Gamma_0) \rceil$
 6. **for** $i \leftarrow 1$ **to** i_{max} **do**
 7. $S \leftarrow$ ThresholdGreedy(\mathcal{R}_1, ϵ)
 8. $\delta_1 = \delta_2 = \delta / (3i_{max})$
 9. compute $f_l(S)$ and $f_u(S^*)$ by (18) and (19)
 10. $\epsilon' \leftarrow f_l(S) / f_u(S^*)$
 11. **if** $\epsilon' \geq (1/2 - 2\epsilon)$ **or** $i == i_{max}$ **then**
 12. **return** S .
 13. $\Gamma_0 \leftarrow 2\Gamma_0$ and generate Γ_0 new random RR sets for $\mathcal{R}_1, \mathcal{R}_2$.
-

6. Experiment

We conducted a number of experiments on the FBIM problem in the course of our work. This section describes the experimental process, including the datasets, the algorithms for comparison, parameter setting, evaluation of the results, and discussion.

6.1. Experiment Settings

6.1.1. Datasets

For a comprehensive experiment, we chose four datasets on SNAP [43]. These datasets have medium to large size and diverse numbers of edges, nodes, and communities. The datasets were the Epinions social network (*Epinions*), the Pokec social network (*Pokec*), the LiveJournal social network and ground-truth communities (*Live-journal*), and the Orkut social network and ground-truth communities (*Orkut*). These datasets have commonly been used to find the seed set with Influence Maximization. Table 2 presents information on the datasets.

Table 2. Statistics of the datasets.

Dataset	#Nodes	#Edges	#Communities	Avg. Degree	Type
Epinions	131,828	841,372	6359	13.4	Directed
Pokec	1,632,803	30,622,564	1284	37.5	Directed
Live-journal	3,997,962	34,681,189	5000	28.5	Directed
Orkut	3,072,441	117,185,083	4745	76.3	Undirected

6.1.2. Environment

We conducted our experiments on a Linux machine with Intel Xeon Gold 6154 (720) @ 3.700GHz CPUs and 3TB RAM. Our implementation was created using g++ 11 and the C/C++ language.

6.1.3. Algorithm Comparison

To the best of our knowledge, there is no existing algorithm that solves the problem of fairness influence maximization for communities under constraints with a pair of lower and upper bound budgets. Therefore, we experimented to compare our three algorithms with OPIMC, as they are similar in terms of implementation. As such, this experiment involves four algorithms, namely, OPIMC, FBIM1, FBIM2, and FBIM3, with different sets of input parameters used to analyze and evaluate their effectiveness. For brevity, we refer to our three proposed algorithms as the FBIM algorithms. The experimental results are shown in Figures 1–4.

In this paper, we do not compare FBIM1 with DSSA, as we showed the efficiency of FBIM1 and compared FBIM1 to DSSA in our previous study [20]. FBIM1 has little advantage compared to FBIM2 and FBIM3 because, as demonstrated by Tang et al. [15], OPIMC is more efficient than DSSA. Nonetheless, we include FBIM1 in this paper as a synthesis of algorithms for our research on the FBIM problem. At the same time, we wanted to compare the performance of FBIM1 with FBIM2, FBIM3, and OPIMC when experimenting with the same set of parameters and data with specific appropriate datasets and parameters, FBIM1 has a conspicuously better influence and coverage ratio than FBIM2, FBIM3, and even OPIMC (see Figures 3 and 4 for the Pokec and Orkut datasets).

As already indicated, these algorithms obtain S and nearly obtain $|S| \simeq k$, allowing $f(S)$ to be maximized. Additionally, whereas OPIMC does not satisfy the fairness constraint, our methods of discovering S do. As such, we contrasted variables such as influence $f(S)$, memory usage, runtime, approximation ratio $f_l(S)/f_u(S^*)$, and coverage ratio S across the target communities. This ratio is the number of communities with elements selected into S that satisfies the constraints of the lower and upper bounds as compared with the original total number of communities (K).

6.1.4. Parameter Setting

We conducted experiments with two sets of parameter settings:

1. We set $k \in [1000, 10000]$, $|C'| = K$ with $K = 20\%k$, the pair $(k_i^l; k_i^u)$ of each C_i to $(0.3; 0.5)$, called $\langle \text{AlgorithmName} \rangle.1$ (such as FBIM1.1, FBIM2.1, FBIM3.1), set $(k_i^l; k_i^u)$ equal to $(0.1; 0.9)$, and called $\langle \text{AlgorithmName} \rangle.2$ similar to case 1. The experiment with these settings is called *Experiment 1*.
2. Here, we used the same k as in *Experiment 1*, with $K = k$. In this case, K is no longer limited to the problem. Because it is already set to the maximum value, that is, in the ultimate case, if the number of communities covered is exactly K (for $K = k$), each community of K communities chooses precisely one element for input S . For the other parameters, the upper bound k_i^u was assigned a fixed value of 0.5 while the lower bound k_i^l varied by 0.1, 0.01 and 0.001, respectively, for $\langle \text{AlgorithmName} \rangle.3$, $\langle \text{AlgorithmName} \rangle.4$, and $\langle \text{AlgorithmName} \rangle.5$. We explain these changes in the discussion of the experimental results. The above case is referred to as *Experiment 2*.

As mentioned above, we executed these algorithms for both information diffusion models, that is, LT and IC. Finally, we used the parameters $\epsilon = 0.1, \delta = \frac{1}{n}$ as the default setting.

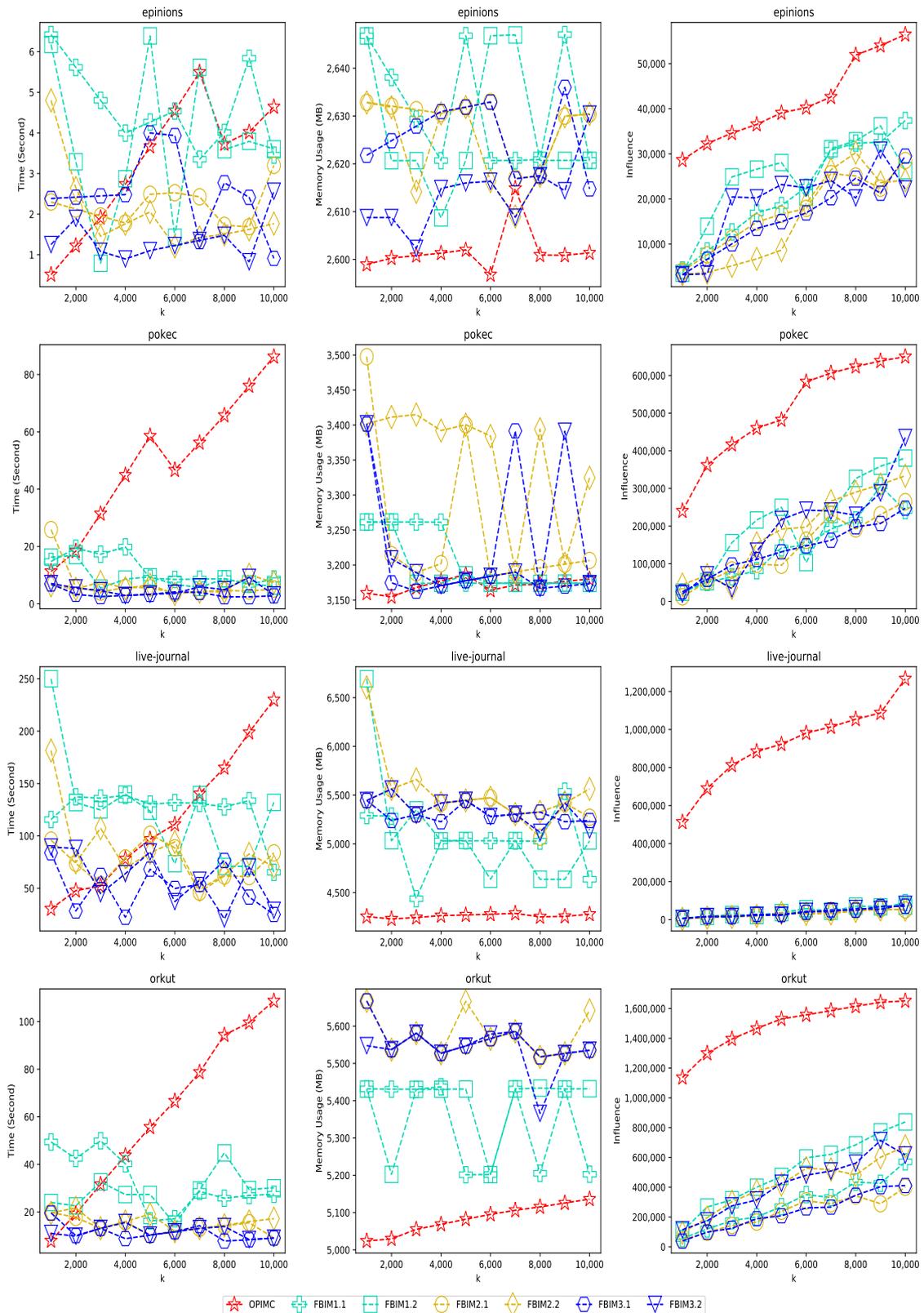


Figure 1. Running time, memory usage, and influence for Experiment 1 under the LT model.

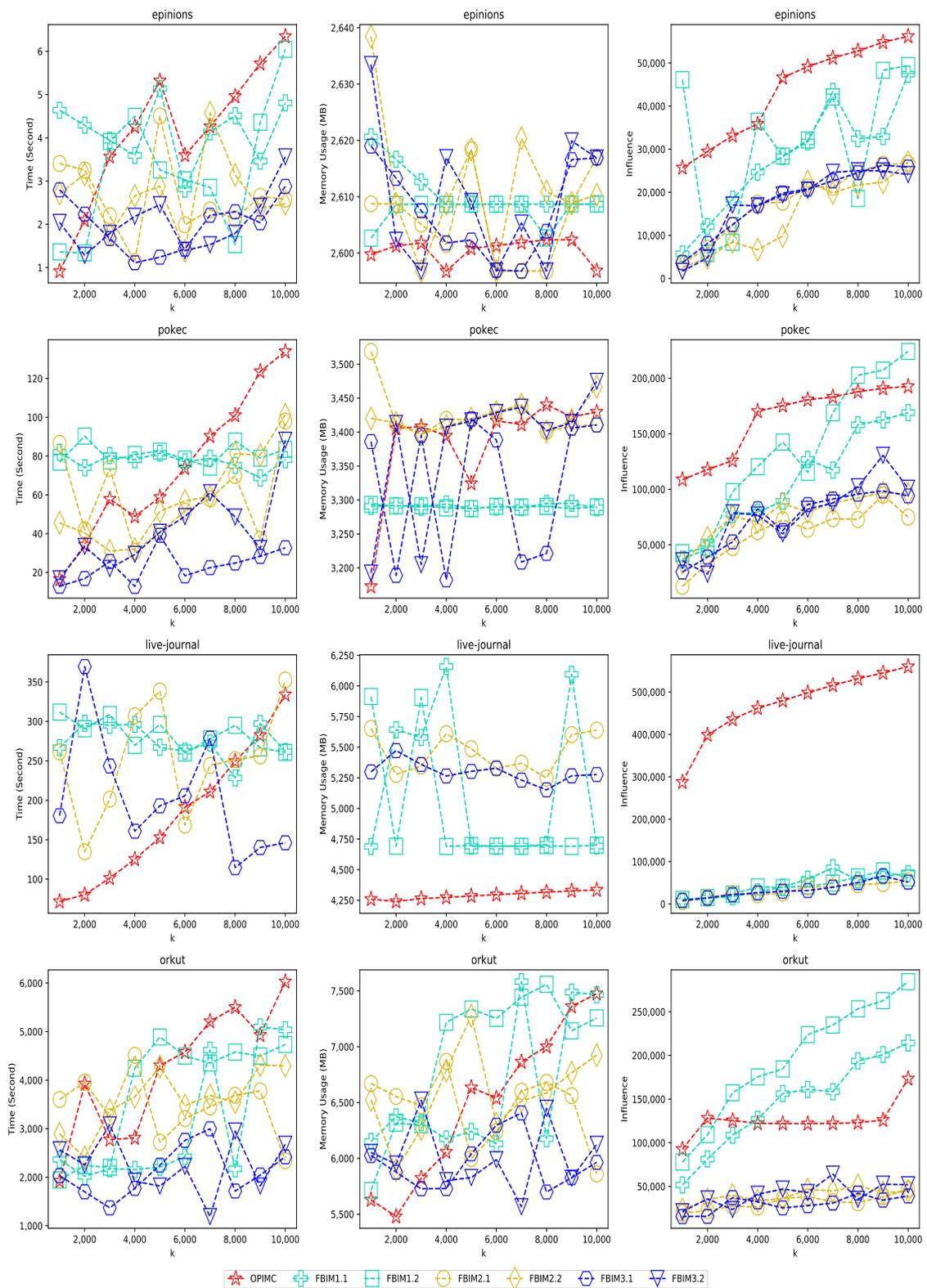


Figure 2. Running time, memory usage, and influence for Experiment 1 under the IC model.

6.2. Discussion and Evaluation of Experimental Results

In this section, we discuss and evaluate the experimental results in order to clarify the strengths and weaknesses of the different algorithms. The results are clearly shown in Figures 1–4.

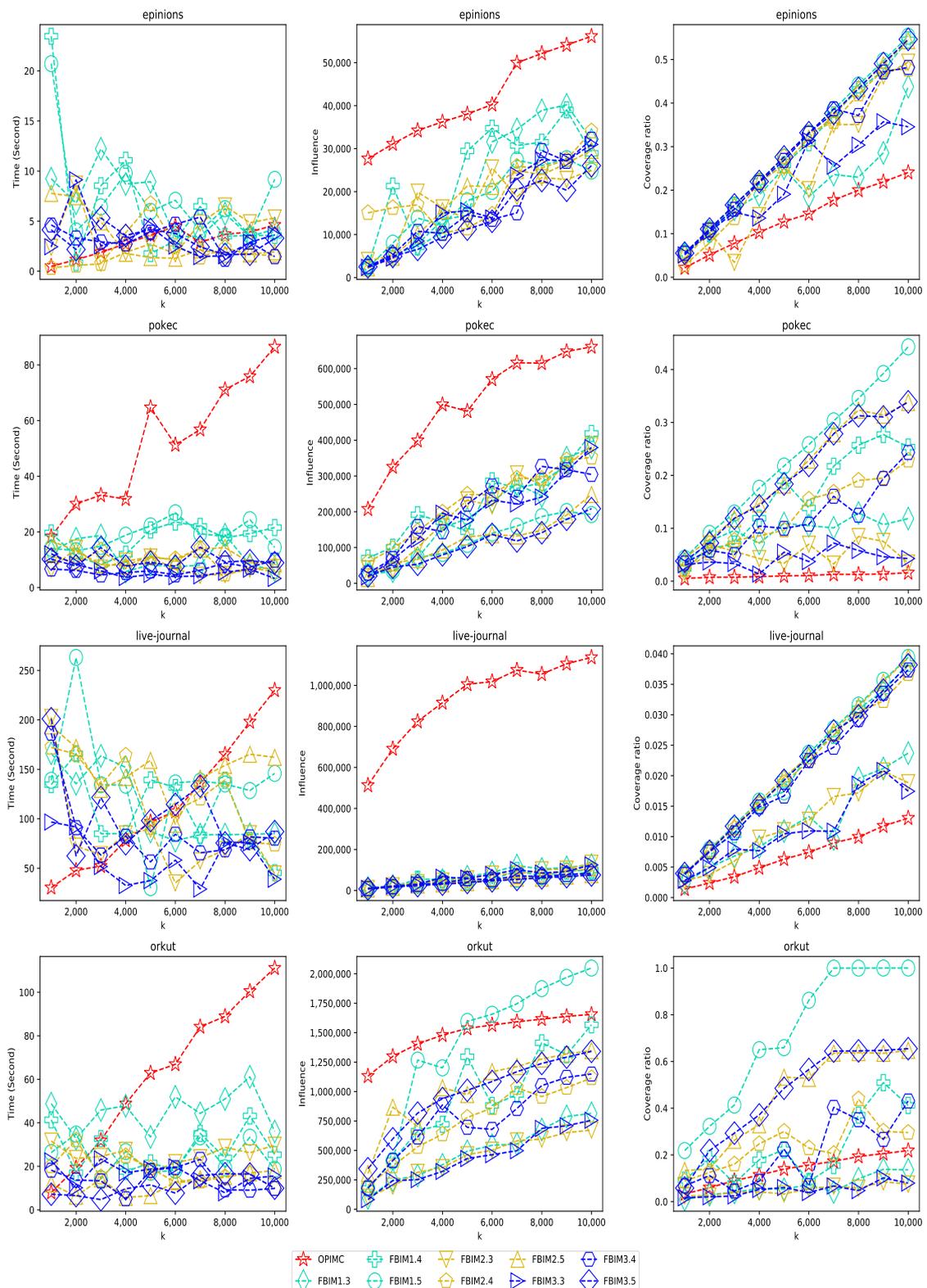


Figure 3. Running time, influence, and coverage ratio for Experiment 2 under the LT model.

6.2.1. Objective Factors Affecting the Efficiency of Algorithms

As clarified in the theoretical analysis of the algorithms, our algorithms guarantee the theoretical optimization probability. However, in the experimental process, a number of objective factors affect the performance of the algorithms.

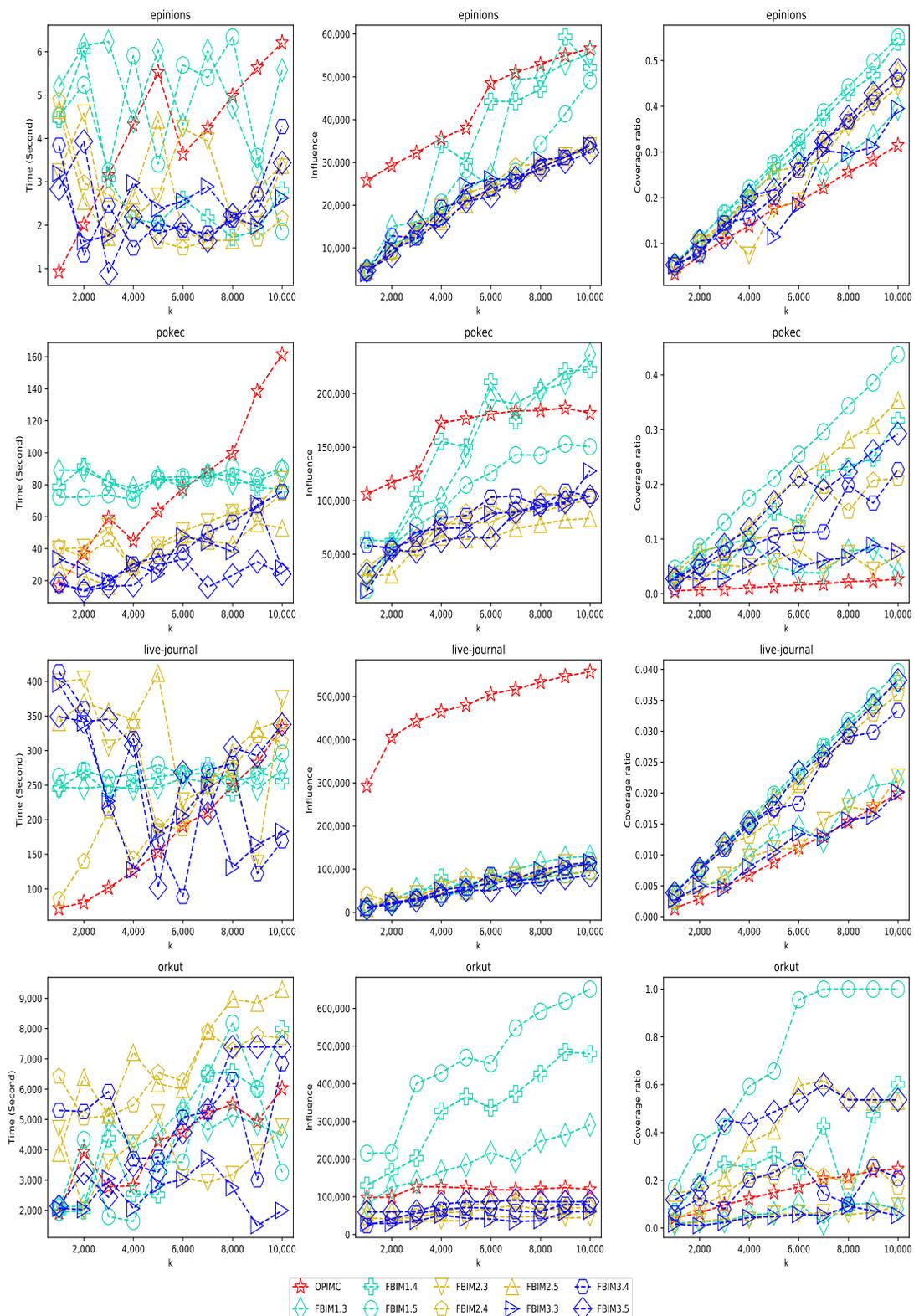


Figure 4. Running time, influence, and coverage ratio for Experiment 2 under the IC model.

1. Data preprocessing

- (a) *Communities detection strategy.* We use the *Directed Louvain* method [44] to detect communities in the datasets. This method applies the idea of simulating the Monte Carlo randomness of the approach. It has been proven that this method is able to achieve a more promising result when extracting communities in

case of a directed graph, although does have flaws. In short, our algorithms' efficiency is affected by the random factor during the communities detection stage.

- (b) *Communities selection strategy.* To ensure objectivity, selecting K communities for the input of the problem FBIM works by choosing randomly, as long as the probability of satisfying the condition for a valid result is within the upper/lower bounds of the fairness constraint. The selection of the combination of communities is entirely random. This selection is repeated many times if the previous combinations do not satisfy the conditions of the FBIM problem. In short, our algorithms are affected by the randomness factor in choosing the initial K communities. However, if we apply these algorithms to real problems, the communities selection stage could depend on a constraint. Thus, a user can improve the selection of elements of S with respect to communities in order to achieve a better result.

2. RIS framework

Our algorithms depend on the generation of sample graphs of the RIS framework. In the first few iterations of the algorithms, if the size of the vertices intersection set of the sample graphs and the selected communities is empty or small, the algorithm must generate more sample graphs to increase this value (i.e., it must increase the number of vertex degrees). After this necessary condition is satisfied, the algorithm finds a seed set S that meets the fairness constraint. Therefore, our algorithms may take extra time to generate sample graphs. Briefly, our algorithms depend on the random factor in creating the sample graph and the number of sample graphs.

In summary, if the above factors are not good, the algorithm takes more time to run before choosing a set S that meets the requirements of the problem. These weaknesses are involved in one of our directions for improvement in future studies.

6.2.2. Experimental Result Evaluation

Experiment 1. For this setting case, the goal of the experiment is to evaluate the algorithm's performance by choosing two narrow and wide bound ranges along with the number of small communities. This setting simulates the general requirement in which it is necessary to select a seed set from several specific finite communities. Experiment 1 evaluates an algorithm's feasibility and performance compared to other algorithms in terms of how disparate they are in runtime and resources.

Hence, Experiment 1 is intended to show how the FBIM algorithms perform in a general setting, including a large range $[k_l^l, k_u^u]$ and a small K . The results in Figures 1 and 2 show that most of the FBIM algorithms run faster than OPIMC; in certain cases, the FBIM algorithms are 4x to 12x faster than OPIMC, especially with increased k . The essence of these algorithms is that OPIMC uses the greedy strategy, which scans all elements on V to find S on the \mathcal{R} set, while, FBIM2 and FBIM3 use an improved greedy approach which scans all elements in communities of \mathcal{C}' ($|\bigcup_{C_i \in \mathcal{C}'} (C_i)| \leq n$). However, there are cases in which the runtime of the FBIM algorithms is extended more than OPIMC, as they are affected by the aforementioned objective factors. Intuitively, the instances of the FBIM algorithms can run longer than OPIMC when k is small. Furthermore, FBIM1 sometimes runs longer than OPIMC because DSSA (the base method of FBIM1) generates more sample graphs than OPIMC. This assertion has been made clear by Tang et al. in [15].

Though the FBIM algorithms typically run faster than OPIMC, their influence on $f(S)$ is less than that of OPIMC. The difference varies by 1.5x to 10x. The reason for this is the small value of K , as when K is small there are fewer communities to cover. This leads to faster calculations while satisfying the fairness constraint. Nevertheless, this is a double-edged sword; because the set of K communities are chosen at random, there is a smaller domain in which to select the seeds, although less time is required to guarantee fairness. Consequently, the algorithm may not always choose nodes with the largest influence in accordance with the improved greedy strategy.

Experiment 2. The goal of this experiment is to evaluate the theoretical potential of the algorithm with the requirement that it cover as many communities as possible; as such, K is not explicitly set, and the lower bound changes while the upper bound remains the same. The lower bound determines the minimum number of elements in the seed set that must be selected from each community to satisfy the requirement that S be extendable. The upper bound is merely a constraint on the maximum number of elements selected from each community.

Thus, for this setting example we want to know the potential value of K to choose a seed set S of size k such that S can cover the majority of the target communities. As a result, we do not specify a value for K . In this case, Figures 3 and 4 show that the lower bound strongly affects the results of the FBIM algorithms. In these figures, the influence of $\langle \text{AlgorithmName} \rangle.4$ and $\langle \text{AlgorithmName} \rangle.5$ increases significantly compared to $\langle \text{AlgorithmName} \rangle.3$; $\langle \text{AlgorithmName} \rangle.3$ has the same lower bound setting as $\langle \text{AlgorithmName} \rangle.2$. The coverage ratio of the FBIM algorithms is greater than that of OPIMC, from 2x to 10x. Even in specific cases, the coverage ratio of OPIMC is only 1.5% (for Pokec), while the ratio of the FBIM algorithms can achieve 33.9–43.9%. For the Orkut dataset, the most extensive dataset of the four, FBIM1 achieves 100%. Furthermore, the value of the coverage ratio and the disparity of these ratios between the FBIM algorithms and OPIMC are proportional to k . In particular, because $\langle \text{AlgorithmName} \rangle.5$ has the smallest lower bound in the lower bounds of the experiments, the algorithm’s coverage ratios are better and the difference distance of the $f(S)$ value from OPIMC is shortened.

Contrarily, set S covers the desired communities, meaning that the affected individuals are the target elements to be affected. In conclusion, the quality of the seed set of the FBIM algorithms can be dealt with by controlling the lower bound k_i^l and k . Nevertheless, lengthy runtime is a disadvantage. The reason for this is that in order to find S , OPIMC only needs to see the first k seeds that meet its requirements, whereas the FBIM algorithms need to find k seeds such that (1) *the same requirements are satisfied as for OPIMC*, and (2) *the requirement that S be extendable is satisfied*. Therefore, the more communities S covers, the more vertex lists must be examined in order to ensure the fairness constraint. This assertion is clearly shown in the experiments with the algorithms in both the LT and IC propagation models.

Considering the memory usage of these algorithms, in two parameters settings under both IC and LT models, the FBIM algorithms almost always require more memory usage, as the datasets contain large communities (especially Epinions Live-journal, and Orkut). This is inevitable due to the FBIM algorithms needing to take an additional step to store and process information about communities. Briefly, although the FBIM algorithms obtain S with an influence spread $f(S)$ smaller than that of OPIMC, their running time is usually faster, and most importantly, they solve the fairness constraint. For the convenience of readers, we summarize the results of our experiments using the FBIM algorithms and OPIMC algorithm in Table 3.

Table 3. Statistical comparison of experimental results of FBIM algorithms vs. OPIMC.

Experiment 1	Runtime	faster than 4x to 12x
	Memory	larger than 1x to 2.7x
	Influence	less than 1.5x to 10x
Experiment 2	Runtime	faster than 0.5x to 6x
	Influence	less than 2x to 6x
	Coverage ratio	greater than 1x to 5x (for Orkut, even FBIM1 can reach 100%; for Pokec, FBIMs can achieve 33.9% → 43.9% while OPIMC achieves 1.5%)

7. Conclusions and Future Work

In this paper, we propose three algorithms to resolve the FBIM problem, which is the FIM problem under a budget threshold constraint and while setting upper and lower bounds to choose seeds in each community in order to ensure fairness. The main result of this study is the finding that our approximation algorithms achieve a $(1/2 - \epsilon)$ -approximation of the optimal solution, and require $O\left(kT \log\left((8 + 2\epsilon)n \frac{\ln \frac{2}{\delta} + \ln \binom{n}{k}}{\epsilon^2}\right)\right)$, $O\left(kT \log \frac{n}{\epsilon^2 k}\right)$, and $O\left(\frac{T}{\epsilon} \log \frac{k}{\epsilon} \log \frac{n}{\epsilon^2 k}\right)$ complexity, respectively. We compare our algorithms with the state-of-the-art OPIMC algorithm by conducting experiments under both the LT and IC information diffusion models. The experiments help to confirm our proven theoretical results. At the same time, we present our algorithms' advantages and disadvantages through the analysis and evaluation of our experimental results. The results indicate that our algorithms are highly scalable and achieve results that satisfy both theoretical assurances and approximation solutions. In addition, these algorithms are feasible and effective even with big data. In future work, we plan to improve the objective factors that affect the efficiency of these algorithms in order to obtain shorter runtime and a more significant influence spread in larger and more varied datasets.

Author Contributions: Conceptualization, B.-N.T.N. and V.-V.L.; Formal analysis, B.-N.T.N.; Investigation, P.N.H.P.; Methodology, B.-N.T.N., P.N.H.P. and V.-V.L.; Project administration, B.-N.T.N.; Resources, B.-N.T.N.; Software, B.-N.T.N. and V.-V.L.; Supervision, V.S.; Validation, V.S.; Writing—original draft, B.-N.T.N.; Writing—review and editing, B.-N.T.N., P.N.H.P., V.-V.L. and V.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Ho Chi Minh city University of Food Industry (HUFI), Ton Duc Thang University (TDTU), and VŠB-Technical University of Ostrava (VŠB-TUO).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All real-world social network datasets used in the experiments can be downloaded at <http://snap.stanford.edu/data/> (accessed on 15 August 2022).

Acknowledgments: The authors are thankful for the support of Ton Duc Thang University (TDTU), Ho Chi Minh City University of Food Industry (HUFI), and VŠB-Technical University of Ostrava (VŠB-TUO).

Conflicts of Interest: The authors declare that they have no competing interests. The study's design, data collection, analysis, and interpretation, the preparation of the paper, and the choice to publish the findings were all made independently of the funders.

References

1. Heidemann, J.; Klier, M.; Probst, F. Online social networks: A survey of a global phenomenon. *Comput. Netw.* **2012**, *56*, 3866–3878. [[CrossRef](#)]
2. Banerjee, S.; Jenamani, M.; Pratihari, D.K. A survey on influence maximization in a social network. *Knowl. Inf. Syst.* **2020**, *62*, 3417–3455. [[CrossRef](#)]
3. Kempe, D.; Kleinberg, J.; Tardos, É. Maximizing the spread of influence through a social network. In Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, 24–27 August 2003; pp. 137–146.
4. Richardson, M.; Domingos, P.M. Mining knowledge-sharing sites for viral marketing. In Proceedings of the Eighth SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton, AB, Canada, 23–26 July 2002; pp. 61–70.
5. Chen, W.; Wang, C.; Wang, Y. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, 24–28 July 2010; pp. 1029–1038.
6. Nguyen, H.T.; Thai, M.T.; Dinh, T.N. Stop-and-stare: Optimal sampling algorithms for viral marketing in billion-scale networks. In Proceedings of the 2016 International Conference on Management of Data, SIGMOD, San Francisco, CA, USA, 26 June–1 July 2016; pp. 695–710.
7. Li, Y.; Fan, J.; Wang, Y.; Tan, K.L. Influence maximization on social graphs: A survey. *IEEE Trans. Knowl. Data Eng.* **2018**, *30*, 1852–1872. [[CrossRef](#)]

8. Kempe, D.; Kleinberg, J.; Tardos, É. Maximizing the spread of influence through a social network. *Theory Comput.* **2015**, *11*, 105–147. [[CrossRef](#)]
9. Banerjee, A.; Chandrasekhar, A.G.; Duflo, E.; Jackson, M.O. The diffusion of microfinance. *Science* **2013**, *341*, 1236498. [[CrossRef](#)] [[PubMed](#)]
10. Yadav, A.; Wilder, B.; Rice, E.; Petering, R.; Craddock, J.; Yoshioka-Maxwell, A.; Hemler, M.; Onasch-Vera, L.; Tambe, M.; Woo, D. Bridging the gap between theory and practice in influence maximization: Raising awareness about hiv among homeless youth. In Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18, Stockholm, Sweden, 13–19 July 2018; pp. 5399–5403.
11. Mirzasoleiman, B.; Babaei, M.; Jalili, M. Immunizing complex networks with limited budget. *EPL (Europhys. Lett.)* **2012**, *98*, 38004. [[CrossRef](#)]
12. Du, N.; Song, L.; Gomez Rodriguez, M.; Zha, H. Scalable influence estimation in continuous-time diffusion networks. In Proceedings of the Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information, Lake Tahoe, NV, USA, 5–8 December 2013; Volume 26.
13. Li, J.; Cai, T.; Mian, A.; Li, R.H.; Sellis, T.; Yu, J.X. Holistic influence maximization for targeted advertisements in spatial social networks. In Proceedings of the 34th International Conference on Data Engineering (ICDE), Paris, France, 16–19 April 2018; pp. 1340–1343.
14. Borgs, C.; Brautbar, M.; Chayes, J.; Lucier, B. Maximizing social influence in nearly optimal time. In Proceedings of the Twenty-Fifth Annual Symposium on Discrete Algorithms, SODA, ACM-SIAM, Portland, OR, USA, 5–7 January 2014; pp. 946–957.
15. Tang, J.; Tang, X.; Xiao, X.; Yuan, J. Online processing algorithms for influence maximization. In Proceedings of the International Conference on Management of Data, SIGMOD '18, Houston, TX, USA, 10–15 June 2018; pp. 991–1005.
16. Pham, C.V.; Ha, D.K.; Vu, Q.C.; Su, A.N.; Hoang, H.X. Influence maximization with priority in online social networks. *Algorithms* **2020**, *13*, 183. [[CrossRef](#)]
17. Sun, G.; Chen, C.-C. Influence maximization algorithm based on reverse reachable set. *Math. Probl. Eng.* **2021**, *2021*, 5535843. [[CrossRef](#)]
18. Khajehnejad, M.; Rezaei, A.A.; Babaei, M.; Hoffmann, J.; Jalili, M.; Weller, A. Adversarial graph embeddings for fair influence maximization over social networks. In Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI, ijcai.org, Yokohama, Japan, 11–17 July 2020; Bessiere, C., Ed.; pp. 4306–4312.
19. Becker, R.; D'Angelo, G.; Ghobadi, S.; Gilbert, H. Fairness in influence maximization through randomization. *J. Artif. Intell. Res.* **2022**, *73*, 1251–1283. [[CrossRef](#)]
20. Nguyen, B.N.T.; Pham, P.N.; Tran, L.H.; Pham, C.V.; Snášel, V. Fairness budget distribution for influence maximization in online social networks. In Proceedings of the 2021 International Conference on Artificial Intelligence and Big Data in Digital Era" (ICABDE 2021), Ho Chi Minh City, Vietnam, 18–19 December 2021.
21. Udawani, R. Multi-objective maximization of monotone submodular functions with cardinality constraint. In Proceedings of the Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems NeurIPS 2018, Montréal, QC, Canada, 3–8 December 2018; Bengio, S., Wallach, H.M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R., Eds.; pp. 9513–9524.
22. Leskovec, J.; Krause, A.; Guestrin, C.; Faloutsos, C.; VanBriesen, J.; Glance, N. Cost-effective outbreak detection in networks. In Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Jose, CA, USA, 12–15 August 2007; pp. 420–429.
23. Chen, W.; Wang, Y.; Yang, S. Efficient influence maximization in social networks. In Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, 28 June–1 July 2009; pp. 199–208.
24. Goyal, A.; Lu, W.; Lakshmanan, L.V. Celf++: Optimizing the greedy algorithm for influence maximization in social networks. In Proceedings of the 20th International Conference Companion on World Wide Web, Hyderabad, India, 28 March–1 April 2011; pp. 47–48.
25. Zhou, C.; Zhang, P.; Zang, W.; Guo, L. On the upper bounds of spread for greedy algorithms in social network influence maximization. *IEEE Trans. Knowl. Data Eng.* **2015**, *27*, 2770–2783. [[CrossRef](#)]
26. Goyal, A.; Lu, W.; Lakshmanan, L.V. Simpath: An efficient algorithm for influence maximization under the linear threshold model. In Proceedings of the IEEE 11th International Conference on Data Mining, Vancouver, BC, Canada, 11–14 December 2011; pp. 211–220.
27. He, X.; Kempe, D. Stability of influence maximization. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 24–27 August 2014; pp. 1256–1265.
28. Liu, Q.; Xiang, B.; Chen, E.; Xiong, H.; Tang, F.; Yu, J.X. Influence maximization over large-scale social networks: A bounded linear approach. In Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, Shanghai, China, 3–7 November 2014; pp. 171–180.
29. Page, L.; Brin, S.; Motwani, R.; Winograd, T. *The Pagerank Citation Ranking: Bringing Order to the Web*; Technical Report 1999-66; Previous Number = SIDL-WP-1999-0120; Stanford InfoLab: Stanford, CA, USA, 1999.
30. Galhotra, S.; Arora, A.; Roy, S. Holistic influence maximization: Combining scalability and efficiency with opinion-aware models. In Proceedings of the 2016 International Conference on Management of Data, Francisco, CA, USA, 26 June–1 July 2016; pp. 743–758.

31. Tang, Y.; Xiao, X.; Shi, Y. Influence maximization: Near-optimal time complexity meets practical efficiency. In Proceedings of the International Conference on Management of Data, SIGMOD, Snowbird, UT, USA, 22–27 June 2014; Dyreson, C.E., Li, F., Özsu, M.T., Eds.; pp. 75–86.
32. Tang, Y.; Shi, Y.; Xiao, X. Influence maximization in near-linear time: A martingale approach. In Proceedings of the 2015 SIGMOD International Conference on Management of Data, Victoria, Australia, 31 May–4 June 2015; Sellis, T.K., Davidson, S.B., Ives, Z.G., Eds.; pp. 1539–1554.
33. Huang, K.; Wang, S.; Bevilacqua, G.; Xiao, X.; Lakshmanan, L.V. Revisiting the stop-and-stare algorithms for influence maximization. *Proc. VLDB Endow.* **2017**, *10*, 913–924. [[CrossRef](#)]
34. Nguyen, H.T.; Dinh, T.N.; Thai, M.T. Revisiting of ‘revisiting the stop-and-stare algorithms for influence maximization’. In *Computational Data and Social Networks*; Chen, X., Sen, A., Li, W.W., Thai, M.T., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 273–285.
35. Tsang, A.; Wilder, B.; Rice, E.; Tambe, M.; Zick, Y. Group-fairness in influence maximization. In Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI, ijcai.org, Macao, China, 10–16 August 2019; Kraus, S., Ed.; pp. 5997–6005.
36. Stoica, A.-A.; Han, J.X.; Chaintreau, A. Seeding network influence in biased networks and the benefits of diversity. In Proceedings of the Web Conference, Taipei, Taiwan, 20–24 April 2020; pp. 2089–2098.
37. El Halabi, M.; Mitrović, S.; Norouzi-Fard, A.; Tardos, J.; Tarnawski, J.M. Fairness in streaming submodular maximization: Algorithms and hardness. In Proceedings of the Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems, Online, 6–12 December 2020; Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H., Eds.
38. Rahmattalabi, A.; Jabbari, S.; Lakkaraju, H.; Vayanos, P.; Izenberg, M.; Brown, R.; Rice, E.; Tambe, M. Fair influence maximization: A welfare optimization approach. In Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence, virtually, 2–9 February 2021; pp. 11630–11638.
39. Ali, J.; Babaei, M.; Chakraborty, A.; Mirzasoleiman, B.; Gummadi, K.; Singla, A. On the fairness of time-critical influence maximization in social networks. In Proceedings of the 2022 IEEE 38th International Conference on Data Engineering (ICDE), Kuala Lumpur, Malaysia, 9–12 May 2022; pp. 1541–1542.
40. Razaghi, B.; Roayaei, M.; Charkari, N. On the Group-Fairness-Aware Influence Maximization in Social Networks. *IEEE Trans. Comput. Soc. Syst.* **2022**. [[CrossRef](#)]
41. Huang, H.; Shen, H.; Meng, Z.; Chang, H.; He, H. Community-based influence maximization for viral marketing. *Appl. Intell.* **2019**, *49*, 2137–2150. [[CrossRef](#)]
42. Badanidiyuru, A.; Vondrák, J. Fast algorithms for maximizing submodular functions. In Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, OR, USA, 5–7 January 2014; pp. 1497–1514.
43. Leskovec, J.; Krevl, A. SNAP Datasets: Stanford Large Network Dataset Collection. Available online: <http://snap.stanford.edu/data> (accessed on 15 August, 2022).
44. Dugué, N.; Perez, A. Directed Louvain: Maximizing Modularity in Directed Networks. Ph.D. Thesis, Université d’Orléans, Orléans, France, 2015.