algorithms

MDPI

Article

Influence Maximization with Priority in Online Social Networks

Canh V. Pham [1], Dung K. T. Ha [1], Quang C. Vu [1,*], Anh N. Su [1] and Huan X. Hoang [2]

[1] Faculty of Information and Security Technology, People's Security Academy, Hanoi 100000, Vietnam; cvpham.cs@gmail.com (C.V.P.); dungha.hvan@gmail.com (D.K.T.H.); anhsungoc@gmail.com (A.N.S.)
[2] Vietnam National University, Hanoi 100000, Vietnam; huanhx@vnu.edu.vn
* Correspondence: quangvc.hvan@gmail.com

Received: 16 June 2020; Accepted: 21 July 2020; Published: 29 July 2020

Abstract: The Influence Maximization (IM) problem, which finds a set of k nodes (called *seedset*) in a social network to initiate the influence spread so that the number of influenced nodes after propagation process is maximized, is an important problem in information propagation and social network analysis. However, previous studies ignored the constraint of priority that led to inefficient seed collections. In some real situations, companies or organizations often prioritize influencing potential users during their influence diffusion campaigns. With a new approach to these existing works, we propose a new problem called *Influence Maximization with Priority* (IMP) which finds out a set seed of k nodes in a social network to be able to influence the largest number of nodes subject to the influence spread to a specific set of nodes U (called *priority set*) at least a given threshold T in this paper. We show that the problem is NP-hard under well-known IC model. To find the solution, we propose two efficient algorithms, called *Integrated Greedy* (IG) and *Integrated Greedy Sampling* (IGS) with provable theoretical guarantees. IG provides a $\left(1-(1-\frac{1}{k})^t\right)$-approximation solution with t is an outcome of algorithm and $t \geq 1$. The worst-case approximation ratio is obtained when $t=1$ and it is equal to $1/k$. In addition, IGS is an efficient randomized approximation algorithm based on sampling method that provides a $\left(1-(1-\frac{1}{k})^t-\epsilon\right)$-approximation solution with probability at least $1-\delta$ with $\epsilon > 0, \delta \in (0,1)$ as input parameters of the problem. We conduct extensive experiments on various real networks to compare our IGS algorithm to the state-of-the-art algorithms in IM problem. The results indicate that our algorithm provides better solutions interns of influence on the priority sets when approximately give twice to ten times higher than threshold T while running time, memory usage and the influence spread also give considerable results compared to the others.

Keywords: social networks; influence maximization with priority; optimization; approximation algorithm

1. Introduction

Presently, Online Social Networks (OSNs) have become an important platform in communication as well as e-commerce. Companies and businesses have leveraged a rapid spread of information thanks to the "word of mouth" effect among friends in social networks as a powerful tool for viral marketing. For instance, companies can provide some ones with free samples over an OSN so that much more people may know about their products and they have more chances to sell them. Influence Maximization (IM) problem [1], a key problem in viral marketing, has been extensively studied for this decade due to its tremendous value in business, viral marketing and influence propagation. Basically, IM aims to find some nodes (called *seedset*) in a social network to inject opinion, innovation or influence that can effect the largest the number of nodes. Kempe et al. [1] first studied

IM as an optimization problem combined with two well-known models, Independent Cascade (IC) and Linear Threshold (LT). Since IM is NP-hard, they designed a native greedy algorithm that returned an $(1 - 1/e)$-approximation solution. The research shows that IM is not only a potential commercial role in viral marketing [2,3] but also a foundation of various applications in many fields such as epidemics control in social network [4–8], social network monitoring [9,10], recommendation system [11], etc. Hence, IM has been extensively studied recently [2,4,12–19].

Although IM has a lot of great applications in viral marketing, previous studies ignored considering the impact on priority users who could play an important role for effectiveness of viral marketing campaigns. In fact, companies often prioritize specific potential customers, who are financially competent or suitable for their products. For examples, if a company produces baby diapers, they tend to introduce the product to married women aged 20 to 45. Supposing that they have some data about user accounts on a social network, hence they launch a promotion with suitable amount of gifts to married female users via this social network. If we only care about the number of influenced individuals, as in the case of IM, we will not evaluate the impact to the potential users and lead to wrong selection of a seed set. Figure 1 shows an example. This network contains 8 nodes and 9 edges, the priority set is $\{b, d\}$ and the weight of each edge (or influence probability) is assigned to 1. Considering the case when the budget $k = 1$ (number of seed nodes), the optimal solution of IM is $\{f\}$ influences to 6 nodes including $\{f, d, g, c, e, h\}$ except b. Hence, IM cannot take effect to all priority nodes.The solution must be $\{a\}$ that has the total influence is only 5.

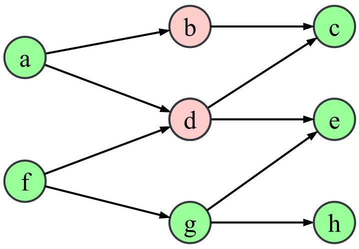


Figure 1. A toy example shows the difference between the influence maximization and our proposed problem.

Motivated by such interesting scenarios, in this paper we investigate the *Influence Maximization with Priority* (IMP) problem, which takes into account the priority constraint for influence process. Given a social network $G = (V, E)$, a *priority set* $U \subset V$, a budget k and a *priority threshold* T, $(T \leq k)$, the goal is to find the seed set S sized at k so that it influences to U at least T and the influence of the cascade is maximized. In fact, IMP is more suitable than IM. Besides, it generalizes IM problem. Nevertheless this problem faces with complicated challenges caused by the constraint of priority. To address this problem, we propose two approximation algorithms, *Integrated Greedy* (IG) and *Integrated Greedy-based Sampling* (IGS), with provable theoretical guarantees. IG meets the theoretical guarantee based on a modification of the natural greedy algorithm while IGS is an efficient randomized approximation algorithm based on sampling method [13–15,20]. This algorithm combines two novel techniques. Firstly, we propose Targeted Reverse Reachable (TRR) concept by modifying the Reverse Reachable Sampling (RR) technique [13–15,20] to estimate influence from a seed set to a given priority set. Secondly, we develop a new strategy to select a set of seeds in accordance with the priority constraint and set the number of samples to give a theoretical guarantees. Because IMP is a separate case of IM, we have built extensive experiments on various real networks to compare our IGS algorithm to the state-of-the-art algorithms for IM problem such as DSSA [15], BCT [2], OPIM about the influence on a given priority set, running time and memory used while the influence spread approximations are ensures as in IM.

Our contributions are summarized as follows:

- We propose the *Influence Maximization with Priority* (IMP) problem that considers *priority constraint* in Influence Maximization (IM) problem. It means we expand the IM by adding a constraint to influence on a given set of users. IMP aims to find the seed set S with size k so that total influence of priority users is at least a given threshold $T, (k \geq T)$ and still maintain the influence of cascade maximized.

- We propose two approximation algorithms, IG and IGS, for the IMP problem. IG algorithm provides an approximation ratio of $\left(1-(1-\frac{1}{k})^t\right)$, where $t \geq k-T$ is an output of the algorithm. In addition, IGS is a randomized approximation algorithm providing an approximation ratio of $\left(1-(1-\frac{1}{k})^t-\epsilon\right)$ with probability at least $1-\delta$, where $\epsilon > 0, \delta \in (0,1)$ are input parameters and t is an output of algorithm.
- We conduct extensive experiments on various real networks such as netHEPT, netPHY, Email-Enron, DBLP, and Twitter ReTweet. The results indicate that our algorithm, IGS, often outperforms state-of-the-art IM algorithms in terms of influence, running time and memory used. In particular, IGS provides the solution which ensures that the influence on the priority set is approximately from twice to 10 times greater than its threshold T while still maintains influence spread approximations as in IM algorithms. Further, we also demonstrate that IGS is faster and uses lower memory than the others in a lot of cases. On the whole, although IGS has to care about how influences to a target given users, IGS still gives considerable fast runtime, low memory used and high maximized influence on all nodes such as state-of-the-art algorithms such as DSSA, BCT, OPIM-C. It proves that IGS has been very well designed.

Related work. Kempe et al.[1] first studied the Influence Maximization (IM) problem inspired by exploiting the influence among users in social networks for viral marketing [21]. They formulated IM as a discrete optimization problem under two classical information diffusion models, Independent Cascade (IC) and Linear Threshold (LT). They proved that IM could be approximated within a ratio of $1-1/e+\epsilon$ for any $\epsilon \in (0,1)$ and proposed a greedy algorithm that provided an approximation ratio of $1-1/e-\epsilon$ for $\epsilon > 0$. Later, Chen et al.[12,16] continued to study IM and proved that to calculate exactly the influence spread of a seeding set was #P-Hard. Hence although many heuristics algorithms have been proposed to solve this problem in large networks, they still have failed to retain the approximation ratio of $1-1/e-\epsilon$ and have provided a low quality solutions such as the cost-effective lazy-forward heuristic (CELF) proposed by Leskovec et al. [22] which is based on improving greedy algorithm to get 700 times faster than the greedy algorithm with Mote-Carlo simulation; a fast heuristics algorithm called PMIA proposed by Chen et al. [12] which constructs a directed acyclic graph to estimate the influence under IC model or the algorithm proposed by the authors in [16] which uses a local directed acyclic graphs (LDAG) to calculate the local influence of nodes under LT model. To keep the $1-1/e-\epsilon$ ratio, research on the approximation approach continues to be explored. Borgs et al. [13] first presented an $(1-1/e-\epsilon)$-approximation algorithm with probability at least $1-\delta$ in $O(kl^2(m+n)\log^2 n/\epsilon^3)$ time complexity by introducing Reverse Influence Sampling (RIS) model. This model has formed the foundation for further algorithm development. [14,15,20,23].

From then on, many works expanded IM in contexts of viral marketing. Nguyen et al. [24] investigated the Budged Influence Maximization (BIM) problem which considered the cost of selecting a node and proposed a $(1-1/\sqrt{e}-\epsilon)$ approximation algorithm. The authors in [2] studied the a generalization of IM and BIM problems, called Cost-aware Targeted Viral Marketing (CTVM). In this work, each node u had an arbitrary cost $c(u)$ and a benefit $b(u)$ and the goal of CTVM was to select a seed set within a given budget so that the total benefit was maximized. We believe that this is the closest problem to our work. In CTVM problem, we can set parameters that maximize the influence on a given target set of users but cannot simultaneously maximize the influence of the others as in our problem. Later, several works improve the approximation as well as the scalability of CTVM algorithms [25,26].

Moreover, there are also many variants of IM problem that were studied. Some works studied the constraints of IM such as [17,18,27], in which edges were associated with a topic influence weight. These problems aimed to find a set of k users that maximized influenced users according to a topic query. However, the proposed algorithms did not provide any theoretical guarantee. Li et al. [28] proposed the Location-aware Influence Maximization (LIM) problem with the goal was to select the k-seed set so that the number of influenced nodes in the given query region was maximized. [29] investigated the Distance-aware Influence Maximization (DAIM) problem which considered the role of distance between users and the promoted location in seed selection. They extended a RIS process model and provided an unbiased estimator for the DAIM problem.

Besides, some works investigated the problem of Competitive Influence Maximization (CIM), which considered the context of IM under the competition of many rivals. Bharathi et al. [30] first formulated the CIM problem under a new competitive propagation model which was an extension of IC model. Chen et al. [12] investigated CIM under the combating with negative opinions based on an assumption that negative information was often more attractive than official information. Some authors considered the problem under many different cases in viral marketing, such as proposing a distance-aware problem [31], expanding the LT model to reflect competition [13,32–34], proposing a heuristic algorithm [35], etc.

Recently, some authors studied the selection of seed nodes in a social network to influence groups of users or communities instead of individuals [36–39]. They argue that in real-world scenarios, creating impact on groups is more beneficial than the individuals in a network. Tsang et al. [36] investigated the Fairness Group Maximization problem with two fairness criteria including maximin fairness and diversity. While the maximin fairness aimed to maximize the minimum influence nodes of any per their population, the criterion of diversity was an alternate fairness concept by extending the notion of individual rationality to group rationality. They proposed an approximation algorithm based on multi-submodular objective function processing techniques. More recent, the authors in [37] proposed exact algorithms for fairness group influence with multiple criteria based on mix integer linear programming formulation on a specific set of sample graphs under IC model. In [38], the authors characterize the intricate relationship between diversity and efficiency, which sometimes may be at odds but may also reinforce each other. Nguyen et al. [39] considered the Influence Maximization problem at the Community level problem, which found seed set of k nodes that influenced to largest number of communities. They showed that the objective function was neither sub-modular nor super-modular and proposed some approximation algorithms with provable guarantees. Different to our studied problem in this paper, these studies did not address the priority set in influence maximization context. Hence the proposed algorithms cannot be applied to the IMP problem.

Organization. The rest of the paper is organized as follows: Section 2 presents information diffusion model and problem definitions. Sections 3 and 4 present our proposed Integrated Greedy and Integrated Greedy-based Sampling algorithms for IMP problem with the theoretical analysis. Experimental results are shown in Section 5. In Section 6 we discuss the future work and conclude this paper.

2. Model and Problem Definition

In this section, we introduce about network model and the well-known Independent Cascade (IC) diffusion information model [1]. Under IC model, we formally define the Influence Maximization with Priority (IMP) problem.

2.1. Graph Notation and Independent Cascade Model

Let $G = (V, E)$ be a directed graph representing a social network with a node set V and a directed edge set E, $|V| = n$ and $|E| = m$. Let $N_{in}(v)$ and $N_{out}(v)$ be two sets of in-neighbors and out-neighbor of a node v, respectively. The notations of S and S^* represent to a seed set that is a solution and an optimal solution of IMP, respectively. We also note $\mathsf{OPT} = \sigma(S^*)$ is the influence of an optimal solution.

In *Independent Cascade* (IC) model, each edge $e = (u, v) \in E$ has an influence probability $p(u, v) \in (0, 1)$ that represents the information transmission from u to v. Each node $v \in V$ has two possible states, *active* and *inactive*. Given a seed set $S \subseteq V$, the diffusion process from S happens in discrete steps $t = 0, 1, \ldots$, as follow:

- At step $t = 0$, all nodes in S is activated.
- At step $t \geq 1$, for an activated node u in previous steps, it has a single chance to activate each inactive neighbour v with the successful probability $p(u, v)$. An activated node remains *active* till the end of the diffusion process.
- The propagation process ends when no more node is activated.

Kempe et al. [1] show that IC model is equivalent to *live-edge* model and estimating the quantity of influence nodes can be done as follows. We first generate a *sample graph* g from original graph G by selecting each edge $e = (u, v) \in E$, independently, with probability $p(u, v)$, and no select edge (u, v) with probability $1 - p(u, v)$. The probability that a realization g can be generated from G (denoted as $g \sim G$) is

$$\Pr[g \sim G] = \prod_{e \in E(g)} p(u, v) \prod_{e \in E \setminus E(g)} (1 - p(u, v)) \tag{1}$$

In this equation, $E(g)$ is the set edge of g. The number of sample graphs is $2^{|E|}$. The influence spread of a seed set S in G is calculated as follows:

$$\sigma(S) = \sum_{g \sim G} \Pr[g \sim G] |R(g, S)| \tag{2}$$

where $R(g, S)$ denotes the set of reachable nodes from S in g. For a set of *priority* nodes U, the influence spread of S to U is calculated as follows:

$$\sigma_U(S) = \sum_{g \sim G} \Pr[g \sim G] |R(g, S \rightarrow U)| \tag{3}$$

where $R_g(S \rightarrow P)$ denotes the set of nodes in U that can reach from S in g. Kempe et al. [1] also show that, $\sigma(\cdot)$ is a monotone and sub-modular function, i.e, for any $A \subset V$, and $v \notin V \setminus B$, we have:

$$\sigma(A + \{v\}) \geq \sigma(A) \tag{4}$$

and for any $A \subseteq B \subset V$, and $v \notin V \setminus B$, we have:

$$\sigma(A + \{v\}) - \sigma(A) \geq \sigma(B + \{v\}) - \sigma(B) \tag{5}$$

We also easy to see that $\sigma_U(\cdot)$ is a monotone and sub-modular function.

2.2. Problem Definition

We investigate Influence Maximization with Priority (IMP) defined as follows:

Definition 1 (IMP problem)**.** *Given a graph $G = (V, E)$ under* IC *model, a positive integer k (budget), the priority set $U \subset V$, and the threshold T with $T \leq k, T \leq |U|$.* IMP *problem asks to find the seed set $S \subset V$, with $|S| \leq k$ and $\sigma_U(S) \geq T$ so that influence spread, $\sigma(S)$, is maximized, i.e, find S that is the solution to the following optimization problem:*

$$\textbf{\textit{maximize:}}\ \ \sigma(S) \tag{6}$$

$$\textbf{\textit{subject to:}}\ \ |S| \leq k \tag{7}$$

$$\sigma_U(S) \geq T \tag{8}$$

IMP becomes IM problem when $U = \emptyset$. Therefore, IM is a special case of IMP and IMP is also NP-hard. In addition, the calculation of the influence function from the seed set is proven to be #P-hard [12]. Thus finding the solution to the problem within the time allowed is very challenging.

3. Integrated Greedy Algorithm

In this section, we first propose Integrated Greedy (IG) Algorithm which is well-known to resolve monotone and sub-modular problems that ensures an lower-bounded of optimization solution. The details of algorithm is described in Algorithm 1.

```
Algorithm 1: Integrated Greedy (IG) algorithm
    Input: Graph $G = (V, E), U \subset V, k, T$
    Output: Seed set $S$, and $t$
 1. $S_1 \leftarrow \emptyset, S_2 \leftarrow \emptyset$
    /* Phase 1:  Greedy strategy for prior set                    */
 2. while $\sigma_U(S_1) < T$ do
 3.     $u \leftarrow \arg\max_{v \in V \setminus S_1} \left(\sigma_U(S_1 \cup \{v\}) - \sigma_U(S_1)\right)$
 4.     $S_1 \leftarrow S_1 \cup \{u\}$
 5. end
 6. $t \leftarrow |k| - |S_1|, i \leftarrow 0$
    /* Phase 2:  Greedy strategy for IM within remain budget      */
 7. while $i < t$ do
 8.     $u \leftarrow \arg\max_{v \in V \setminus S_2} \left(\sigma(S_2 \cup \{v\}) - \sigma(S_2)\right)$
 9.     if $u \in S_1$ then
10.         $t \leftarrow t + 1$
11.     end
12.     $S_2 \leftarrow S_2 \cup \{u\}, i \leftarrow i + 1$
13. end
14. $S \leftarrow S_1 \cup S_2$
15. return $S, t$;
```

Assume S_1 is the solution of the problem that finds the minimum seed nodes such that the influence on the priority set is greater than threshold T, and S_2 is a solution of IM problem. The main idea of this algorithm is to modify the native greedy algorithm [1] by combining two above solutions.

The algorithm is divided into two main phases. In the first phase, it tries to find a solution S_1 by a greedy strategy (line 2–4). In each iterator, the algorithm chooses a node u with largest *influence incremental* to set U into S_1 (line 3-4) until the $\sigma_U(S_1) \geq T$. Since $T < k$, $|S_1| \leq T < k$. Denote $t = k - T$ as the remaining budget (line 6). The algorithm next finds the candidate solution S_2 for IM with the remaining budget t by using a greedy method in the second phase (line 6-10). In each iterator i, it selects a node u with largest *influence incremental* (line 7). If u already belongs to S_1, the algorithm increases t by 1 (line 8–9). This phase ends when the remaining budget t is exhausted (line 6). Finally, the algorithm returns the solution S which unites S_1 and S_2. It is easy to confirm that $|S| = k$, and $t > T - k \geq 1$ since $k > T$. Theorem 1 shows the approximation guarantee of IG algorithm.

Theorem 1. IG *algorithm returns (S, t), where S is a feasible solution and $t \geq 1$, satisfies:*

$$\sigma(S) \geq \left(1 - \left(1 - \frac{1}{k}\right)^t\right) \sigma(S^*)$$

The worst-case approximation ratio is obtained when t = 1 and it is equal to 1/k.

Proof. Denote $S^*_{\mathsf{IM}} = \{s_1, s_2, \ldots, s_k\}$ is an optimal solution of IM problem for input data of Algorithm 1 (the graph G and budget k). Obviously, we have $\sigma(S^*_{\mathsf{IM}}) \geq \sigma(S^*)$. After ending the second phase, assume that $S_2 = \{u^1_2, u^2_2, \ldots, u^t_2\}$, $S^i_2 = \{u^1_2, u^2_2, \ldots, u^i_2\}$, and $S^0_2 = \emptyset$. In the second phase, the algorithm repeatedly selects a node u of which incremental influence gain is largest and due to the function $\sigma(\cdot)$ is monotone and sub-modular [1], so we have:

$$\sigma(S^*_{\mathsf{IM}}) - \sigma(S^i_2) \leq \sigma(S^*_{\mathsf{IM}} \cup S^i_2) - \sigma(S^i_2) \tag{9}$$

$$\leq \sum_{j=1}^{k} \Big(\sigma(S^i_2 \cup \{s_1, s_2, \ldots, s_j\}) - \sigma(S^i_2 \cup \{s_1, s_2, \ldots, s_{j-1}\}) \Big) \tag{10}$$

$$\leq \sum_{j=1}^{k} \Big(\sigma(S^i_2 \cup \{s_j\}) - \sigma(S^i_2) \Big) \text{ (Due to } \sigma \text{ is a sub-modular function)} \tag{11}$$

$$\leq k \cdot \max_{s \in S^*_{\mathsf{IM}}} \Big(\sigma(S^i_2 \cup \{s\}) - \sigma(S^i_2) \Big) \tag{12}$$

$$\leq k \cdot \Big(\sigma(S^{i+1}_2) - \sigma(S^i_2) \Big) \tag{13}$$

Therefore, for any $i = 0, \ldots, t-1$, we have

$$\sigma(S^{i+1}_2) - \sigma(S^i_2) \geq \frac{1}{k}(\sigma(S^*_{\mathsf{IM}}) - \sigma(S^i_2)) \tag{14}$$

Minus two inequality terms to $\sigma(S^*_{\mathsf{IM}})$, we have:

$$\sigma(S^{i+1}_2) - \sigma(S^i_2) - \sigma(S^*_{\mathsf{IM}}) \geq \frac{1}{k}\sigma(S^*_{\mathsf{IM}}) - \sigma(S^*_{\mathsf{IM}}) - \frac{1}{k}\sigma(S^i_2) \tag{15}$$

Rearrange the terms of the above inequality, we have

$$\sigma(S^{i+1}_2) - \sigma(S^*_{\mathsf{IM}}) \geq \left(1 - \frac{1}{k}\right)(\sigma(S^i_2) - \sigma(S^*_{\mathsf{IM}})) \tag{16}$$

$$\geq \left(1 - \frac{1}{k}\right)^t (\sigma(S^0_2) - \sigma(S^*_{\mathsf{IM}})) \tag{17}$$

Together with the fact that $S^0_2 = \emptyset$ and $\sigma(\emptyset) = 0$, the above inequality implies

$$\sigma(S_2) = \sigma(S^t_2) \geq \left(1 - \left(1 - \frac{1}{k}\right)^t\right)\sigma(S^*_{\mathsf{IM}}) \tag{18}$$

Since $\sigma_U(S_1) \geq T$ and $S = S_1 \cup S_2$, S is feasible solution of IMP, and

$$\sigma(S) \geq \sigma(S^t_2) \geq \left(1 - \left(1 - \frac{1}{k}\right)^t\right)\sigma(S^*_{\mathsf{IM}}) \geq \left(1 - \left(1 - \frac{1}{k}\right)^t\right)\sigma(S^*) \tag{19}$$

which proves the theorem! □

Although Algorithm 1 can provide an approximation guarantee, but it cannot work with real-social networks because the calculation of the influence function $\sigma(S)$ is #P-hard under IC model [12]. To overcome this challenge, we propose a randomize algorithm with provable approximation guarantee based on combining IG with a sampling technique.

4. Sampling Algorithm with Provable Guarantees

In this section, we present an efficient algorithm for IMP problem called *Integrated Greedy Sampling* (IGS) algorithm that can provide an guarantee theoretical. In addition, we show that our algorithm can also be applied to large networks in experiments.

4.1. Estimator of Influence Functions

Firstly, we recap the concept of *Reachable Reverse (RR)* set [40] to estimate influence function $\sigma(\cdot)$. Base on that, we propose the concept of *Targeted Reachable Reverse (TRR)* set to estimate influence function $\sigma_U(S)$. Then we propose IGS algorithm and provide theoretical analysis based on statistical evidence.

Definition 2 (Reachable Reverse (RR) set [40])**.** *Given a graph $G = (V, E)$ under* IC *model. A random RR set R_j is generated from G by:*

1. *Picking a source node u with probability $\frac{1}{n}$.*
2. *Generating a sample graph g from G, and returning R_j as nodes which can be reached from u in g.*

For a random RR set R_j, define a random variable $X_g(S) = \min\{1, |R_g \cap S|\}$. Borgs et al. [40] show that RR samples can be used to estimate the influence function by applying the following Lemma.

Lemma 1. *For any set of nodes $S \subseteq V$, we have $\sigma(S) = n \cdot \mathbb{E}[X_g(S)]$.*

Given a set of RR set $\mathcal{R}$, and a set node S, we can approximate the value of $\sigma(S)$ by $\hat{\sigma}(S)$ defined as follow:

$$\hat{\sigma}(S) = \frac{n}{|\mathcal{R}|} \sum_{R_g \in \mathcal{R}} X_g(S) \tag{20}$$

Generating RR sets can be accomplished by using IM algorithms in [13–15,20,23]. The common algorithm for generating RR set R_j is described in Algorithm 2. This algorithm first selects a source node u with a probability $\frac{1}{n}$ to add into R_j. The algorithm uses a queue Q to store the visited nodes. Initially, u is also added to Q. The algorithm next retrieves each node v in Q and picks an incoming node x with probability $p(x, v)$ (line 6). If successful, it adds x in to Q and R_j. This process takes place until the set Q is empty.

Algorithm 2: Generating RR sample under IC model

```
   Input: Graph $G = (V, E)$ under IC model
   Output: A RR Sample $R_j$
1. Pick a source node $u$ with probability $\frac{1}{n}$
2. Initialize a queue $Q = \{u\}$ and $R_j = u$
3. while $Q$ is not empty do
4.     $v \leftarrow Q.pop()$
5.     foreach $x \in N_{in}(v) \setminus (R_j \cup Q)$ do
6.         With probability $p(x, v)$: $Q.push(x)$ and $R_j \leftarrow R_j \cup \{u\}$
7.     end
8. end
9. return $R_j$
```

We now introduce the definition of *Targeted Reachable Reverse (TRR)* Set on the basis of modifying RR concept.

Definition 3 (Targeted Reachable Reverse (TRR) Set)**.** *Given a graph $G = (V, E)$ under* IC *model. A random TRR set R_j^U is generated from G by:*

1. *Picking a source node $u \in U$ with probability $\frac{1}{|U|}$.*
2. *Generating a sample graph g from G, and returning R_g^U as nodes which can be reached from u in g.*

We define a random variable $Y_g(A) = \min\{1, |R_g^U \cap S|\}$. Similar to Lemma 1, Lemma 2 shows that we can use the value of $Y_g(S)$ to estimate function $\sigma_U(S)$.

Lemma 2. *For any set of nodes $S \subseteq V$, we have $\sigma_U(S) = |U| \cdot \mathbb{E}[Y_g(S)]$*

Proof. Denote $R_g^U(u)$ is a TRR sample with a source node u for the sample graph g, we have:

$$\begin{aligned}\sigma_U(S) &= \sum_{g\sim G} |R(g, S \to U)| \\ &= \sum_{u\in U}\sum_{g\sim G} \Pr[g \sim G][\exists v \in S : u \text{ is reached from } v] \\ &= \sum_{u\in U}\sum_{g\sim G} \Pr[g \sim G][\exists v \in S : v \in R_g^U(u)] \\ &= |U| \sum_{u\in V} \frac{1}{|U|} \sum_{g\sim G} \Pr[g \sim G] Y_g(S) \\ &= |U| \sum_{u\in V}\sum_{g\sim G} \Pr[u \text{ is source node}] \Pr[g \sim G] Y_g(S) \\ &= |U| \cdot \mathbb{E}[Y_g(S)]\end{aligned}$$

The transition from the second equality to the third equality comes from the definition of $R_g^U(u)$ and from the third to the fourth then to the fifth is caused by the distribution of choosing a node u as a source node. □

Given a set of TRR samples $\mathcal{R}$ and a set node S, we define and an approximation value of $\sigma_U(S)$ as follow:

$$\hat{\sigma}_U(S) = \frac{|U|}{|\mathcal{R}|} \sum_{R_g^U \in \mathcal{R}} Y_g(S) \tag{21}$$

From Lemma 2, we can give a good approximation of $\sigma_U(\cdot)$ when the number of TRR samples is large enough. We can re-use Algorithm 2 to generate a TRR set R_j^U by a modification. We replace line 1 in the algorithm by picking source node $u \in U$ with probability $\frac{1}{|U|}$ and leave the rest as is.

4.2. Algorithm Description and Theoretical Analysis

Algorithm description. The algorithm is detailed in Algorithm 3. It generates the set of N_U TRR sets $\mathcal{R}_1$, and set two candidate solutions S_1, S_2 empty at first. Then the body of the algorithm divides into two phases. In phase 1, it finds a candidate solution S_1 with minimum-size so that $\hat{\sigma}(S) \geq (1 + \alpha)T$ by using a greedy strategy with potential function $\hat{\sigma}$ over $\mathcal{R}_1$. In each iterator, it selects a node u with maximal incremental value of the potential function (line 4) until $\hat{\sigma}(S) \geq (1 + \alpha)T$. The candidate solution S_1 obtained by this phase satisfies the priority constraint, $\sigma_U(S_1) \geq T$ with probability at least $1 - \delta$ (Lemma 4).

The phase 2 selects a candidate solution S_2 with the remaining budget ($t = k - |S_1|$) so that the influence spread $\sigma(\cdot)$ is maximized. In this phase, it first sets the parameters ϵ_1, t_{max}, N_{max} and generates N_1 set of RR samples $\mathcal{R}_2$. The main of this phase operates in several iterators (line 12-27) until meeting the stopping condition (line 22). In each iterator, it finds a candidate solution S_2 by a greedy strategy. It picks a node u with maximal incremental of approximation influence $\hat{\sigma}(\cdot)$ over $\mathcal{R}_2$ (line 12) until t nodes are selected. Similar to IG algorithm, if u already belongs to S_1, the algorithm increases t by 1. After that, the algorithm checks the quality of candidate solution S_2 (line 17). It calculates $F_l(S_2, \mathcal{R}_2, \delta)$- a lower-bounded of $\sigma(S_2)$, and $F_u(S_2, \mathcal{R}_2, \delta)$-an upper-bounded of an optimal solution respect to IMP problem. These functions ensure the statistical criterion, which are claimed in the Lemmas 5 and 6. If solution S_2 meets the approximation condition (line 19), the algorithm returns S_2. If not, it moves to the next iterator and stops when the number of TRR samples is at least N_{max} (line 21).

Algorithm 3: Integrated Greedy -based Sampling (IGS) algorithm

```
    Input: Graph $G = (V, E), U \subset V, k, T, \epsilon, \alpha, \delta \in (0, 1)$
    Output: Seed set $S$
1.  Generate a set of $N_U = (2 + \frac{2}{3}\alpha)|U| \frac{\ln(\binom{|U|}{\lfloor |U|/2 \rfloor}/\delta)}{(T + T\alpha)\alpha^2}$ TRIS sets $\mathcal{R}_1$.
2.  $S_1 \leftarrow \emptyset, S_2 \leftarrow \emptyset$
    /* Phase 1                                                    */
3.  while $\hat{\sigma}_U(S_1) < T + \alpha T$ do
4.      $u \leftarrow \arg\max_{v \in V \setminus S_1} \left(\hat{\sigma}_U(S_1 \cup \{v\}) - \hat{\sigma}_U(S_1)\right)$
5.      $S_1 \leftarrow S_1 \cup \{u\}$
6.  end
    /* Phase 2                                                    */
7.  $\epsilon_1 \leftarrow \frac{\epsilon}{2(1-1/e)-\epsilon}$
8.  $t_0 \leftarrow k - |S_1|, \delta_1 \leftarrow \frac{\delta}{6}, t_{max} \leftarrow \arg\max_{j \in \{t, t+1, t+2, \dots, k\}} \ln(\binom{n}{j}/\delta_1)/j$
9.  $N_1 \leftarrow \frac{\ln(1/\delta_1)}{\epsilon_1^2}, N_{max} \leftarrow \frac{(2+\frac{2}{3}\epsilon_1) n \ln(\binom{n}{t_{max}}/\delta_1)}{t_0 \epsilon_1^2}$
10. $i_{max} = \lceil \frac{N_{max}}{N_1} \rceil, \delta_2 \leftarrow \frac{\delta}{3 i_{max}}$
11. Generate set of $N_1$ RR samples $\mathcal{R}_2$
12. repeat
13.     $t \leftarrow t_0, i \leftarrow 0$
14.     while $i < t$ do
15.         $u \leftarrow \arg\max_{v \in V \setminus S_2} \left(\hat{\sigma}(S_2 \cup \{v\}) - \hat{\sigma}(S_2)\right)$
16.         if $u \in S_1$ then
17.             $t \leftarrow t + 1$
18.         end
19.         $S_2 \leftarrow S_2 \cup \{u\}, i \leftarrow i + 1$
20.     end
21.     Calculate $F_l(S_2, \mathcal{R}_2, \delta_2)$ and $F_u(S_2, \mathcal{R}_2, \delta_2)$
22.     if $\frac{F_l(S_2, \mathcal{R}_2, \delta_2)}{F_u(S_2, \mathcal{R}_2, \delta_2)} \geq 1 - (1 - \frac{1}{k})^t - \epsilon$ then
23.         return $S_2$
24.     else
25.         Generate $|\mathcal{R}_2|$ RR samples and add them into $\mathcal{R}_2$
26.     end
27. until $|\mathcal{R}_2| \geq N_{max}$;
28. $S \leftarrow S_1 \cup S_2$
29. return $S$;
```

Theoretical analysis. Fortunately, the sequence of random variables $X_g(S)$ and $Y_g(S)$ constructed from the RR and TRR samples can be shown to form a martingale. For any random variable $X_g(S) \in [0,1]$, let a random variable $M_i = \sum_{j=1}^{i}(X_g^i(S) - \mu), \forall i \geq 1$, where $\mu = \mathbb{E}[X_g]$. For a sequence of random variables $M_1, M_2, \ldots$ we have $\mathbb{E}[M_i|M_1, \ldots, M_{j-1}] = \mathbb{E}[M_{i-1}] + \mathbb{E}[X_g^i(S) - \mu] = \mathbb{E}[M_{i-1}]$. Hence, $M_1, M_2, \ldots$ be a form of martingale [41]. Similarly, Y_g is also a form of martingale. Therefore, the following concentration inequality [41] applies:

Lemma 3. *If $M_1, M_2, \ldots$ be a form of martingale, $|M_1| \leq a$, $|M_j - M_{j-1}| \leq a$ for $j \in [1, i]$, and*

$$\mathsf{Var}[M_1] + \sum_{j=2}^{i} \mathsf{Var}[M_j|M_1, M_2, \ldots, M_{j-1}] = b \tag{22}$$

where $\mathsf{Var}[\cdot]$ denotes the variance of a random variable. Then, for any λ, we have:

$$\Pr[M_i - \mathbb{E}[M_i] \geq \lambda] \leq \exp\left(-\frac{\lambda^2}{\frac{2}{3}a\lambda + 2b}\right) \tag{23}$$

Apply this Lemma with $|M_1| = |X_g^1(S)| \leq 1$, $|M_j - M_{j-1}| = |X_g^j(S) - X_g^{j-1}(S)| \leq 1$, $\mathsf{Var}[M_1] = \mathsf{Var}[X_g^1(S) - \mu] = \mathsf{Var}[X_g(S)]$, $\mathsf{Var}[M_j|M_1, M_2, \ldots, M_{j-1}] = \mathsf{Var}[X_g^j(S) - \mu] = \mathsf{Var}[X_g(S)]$, and $\mathsf{Var}[X_g(S)] \leq \mu(1-\mu) \leq \mu$, we have:

$$\Pr\left[\sum_{i=1}^{|\mathcal{R}|} X_g^i(S) - |\mathcal{R}| \cdot \mu \geq \lambda\right] \leq \exp\left(-\frac{\lambda^2}{\frac{2}{3}\lambda + 2\mu|\mathcal{R}|}\right) \tag{24}$$

Similarly, $-M_1, \ldots, -M_i, \ldots$ also form a Martingale, so apply Lemma 3, we have:

$$Pr\left[\sum_{i=1}^{|\mathcal{R}|} X_g^i(S) - |\mathcal{R}| \cdot \mu \leq -\lambda\right] \leq \exp\left(-\frac{\lambda^2}{2\mu|\mathcal{R}|}\right) \tag{25}$$

Let $\lambda = \epsilon\mu|\mathcal{R}|$ and put it in two above inequalities, we have:

$$\Pr[\sum_{i=1}^{|\mathcal{R}|} X_g^i - |\mathcal{R}| \cdot \mu \geq \epsilon|\mathcal{R}|\mu] \leq \exp\left(-\frac{\epsilon^2|\mathcal{R}|\mu}{2 + \frac{2}{3}\epsilon}\right) \tag{26}$$

$$\Pr[\sum_{i=1}^{|\mathcal{R}|} X_g^i - |\mathcal{R}| \cdot \mu \leq -\epsilon|\mathcal{R}|\mu] \leq \exp\left(-\frac{\epsilon^2|\mathcal{R}|\mu}{2}\right) \tag{27}$$

The following Lemma shows the lower-bound of the influence of candidate solution S_1.

Lemma 4. *The candidate solution S_1 obtained by phase 1 of Algorithm 3 satisfies* $\Pr[\sigma_U(S_1) \geq T] \geq 1 - \delta$

Proof. Denote $\mu_Y = \mathbb{E}[Y_g] = \frac{\sigma_U(S_1)}{|U|}$, and $\hat{\mu}_Y = \frac{1}{N_U}\sum_{i=1}^{N_U} Y_g^i = \frac{\hat{\sigma}_U(S_1)}{|U|} \geq \frac{(T+\alpha T)}{|U|}$. Apply (27) for set $\mathcal{R}_1$, we have:

$$\begin{aligned}\Pr[\hat{\mu}_Y \leq (1-\alpha)\mu_Y] &= \Pr\left(\sum Y_g^i - N_U\mu_Y \leq -\alpha N_U \mu_Y\right)\\ &\leq \exp\left(\frac{-\epsilon^2 N_U \mu_Y}{2}\right)\\ &\leq \exp\left(\frac{-\epsilon^2 \hat{\mu}_Y N_U}{2(1-\alpha)}\right)\\ &\leq \exp\left(\frac{-\epsilon^2(T+\alpha T)}{2(1-\alpha)|U|}N_U\right)\\ &\leq \exp\left(\frac{-(2+\frac{2}{3}\alpha)\ln(\binom{|U|}{\lfloor |U|/2\rfloor}/\delta)}{2(1-\alpha)}\right)\\ &\leq \exp\left(-\ln(\binom{|U|}{\lfloor |U|/2\rfloor}/\delta)\right) \leq \frac{\delta}{\lfloor |U|/2\rfloor}\end{aligned}$$

We assume that the event $\hat{\mu}_Y \leq (1-\alpha)\mu_Y$ happens, apply (26) for set $\mathcal{R}_1$, we have:

$$\begin{aligned}\Pr[\sigma_U(S_1) \leq T] &\leq \Pr\left(\sigma_U(S_1) \leq \frac{\hat{\sigma}_U(S_1)}{1+\alpha}\right) &(28)\\ &= \Pr\left(\hat{\sigma}_U(S_1) \geq (1+\alpha)\sigma_U(S_1)\right) &(29)\\ &= \Pr\left(\frac{|U|}{N_U}\sum_{i=1}^{N_U} Y_g^i - |U|\mu_Y \geq |U|\alpha\mu_Y\right) &(30)\\ &= \Pr\left(\sum_{i=1}^{N_U} Y_g^i - N_U\mu_Y \geq N_U\alpha\mu_Y\right) &(31)\\ &\leq \exp\left(-\frac{\alpha^2\mu_Y}{2+\frac{2}{3}\alpha}N_U\right) &(32)\\ &\leq \exp\left(-\frac{\alpha^2\hat{\mu}_Y}{(2+\frac{2}{3}\alpha)(1-\alpha)}N_U\right) &(33)\\ &\leq \exp\left(-\frac{\alpha^2\hat{\sigma}_U(S_1)}{(2+\frac{2}{3}\alpha)(1-\alpha)(T+\alpha T)}N_U\right) &(34)\\ &\leq \exp\left(-\ln(\binom{|U|}{\lfloor |U|/2\rfloor}/\delta)\right) &(35)\\ &\leq \frac{\delta}{\binom{|U|}{\lfloor |U|/2\rfloor}} &(36)\end{aligned}$$

Assume that $|S_1| = k_1$, there are at most $\binom{n}{k_1}$ possibilities for the candidate solutions S_1. Therefore,

$$\Pr[\exists S_1 : \sigma_U(S_1) \leq T] \leq \binom{n}{k_1}\frac{\delta}{\binom{|U|}{\lfloor |U|/2\rfloor}} \leq \delta \tag{37}$$

□

Lemma 5 (Lower-bound)**.** *For any $\delta \in (0,1)$, a set of RR samples $\mathcal{R}$, let $c = \ln(\frac{1}{\delta})$, and*

$$F_l(\mathcal{R}, S, \delta) = \min\left\{\hat{\sigma}(S) - \frac{nc}{3|\mathcal{R}|}, \hat{\sigma}(S) + \frac{n}{|\mathcal{R}|}\left(\frac{2c}{3} - \sqrt{\frac{4c^2}{9} + 2|\mathcal{R}|c\frac{\hat{\sigma}(S)}{n}}\right)\right\} \tag{38}$$

We have $\Pr[\sigma(S) \geq F_l(\mathcal{R}, \delta)] \geq 1 - \delta$.

Proof. Denote $\mu = \mathbb{E}[X_g(S)] = \frac{\sigma(S)}{n}$ and $\hat{\mu} = \frac{1}{n}\sum_{R_g \in \mathcal{R}} X_g(S) = \frac{\hat{\sigma}(S)}{n}$. Apply (24) with $\lambda = \frac{c}{3} + \sqrt{\frac{c^2}{9} + 2c\mu|\mathcal{R}|}$, we have:

$$\Pr\left[\sum_{j=1}^{T} X_g(S) - |\mathcal{R}| \cdot \mu \geq \lambda\right] \leq \delta \tag{39}$$

Therefore, the following event happens with probability at least $1 - \delta$

$$\sum_{j=1}^{T} Z_j(S) - |\mathcal{R}| \cdot \mu \leq \lambda \Leftrightarrow |\mathcal{R}|\hat{\mu} - |\mathcal{R}|\mu - \frac{c}{3} \leq \sqrt{\frac{c^2}{9} + 2c\mu|\mathcal{R}|} \tag{40}$$

We consider two following cases:

Case 1: If $|\mathcal{R}|\hat{\mu} - |\mathcal{R}|\mu - \frac{c}{3} \leq 0$, then $\mu \geq \hat{\mu} - \frac{c}{3|\mathcal{R}|}$.
Case 2: if $|\mathcal{R}|\hat{\mu} - |\mathcal{R}|\mu - \frac{c}{3} > 0$, (40) becomes:

$$\left(|\mathcal{R}|\hat{\mu} - |\mathcal{R}|\mu - \frac{c}{3}\right)^2 \leq \frac{c^2}{9} + 2c\mu|\mathcal{R}| \tag{41}$$

$$\Leftrightarrow (\hat{\mu} - \mu)^2|\mathcal{R}| + \frac{4c}{3}(\hat{\mu} - \mu) - 2c\hat{\mu} \leq 0 \tag{42}$$

Solve the above inequality for μ, we obtain:

$$\mu \geq \hat{\mu} + \frac{1}{T}\left(\frac{2c}{3} - \sqrt{\frac{4c^2}{9} + 2|\mathcal{R}|c\hat{\mu}}\right) \tag{43}$$

Combine two above cases and replace $\mu = \frac{\sigma(S)}{n}, \hat{\mu} = \frac{\hat{\sigma}(S)}{n}$, we obtain the proof. □

Lemma 6 (Upper-bound). *For any* $\delta \in (0, 1)$, *in an iterator* t *of Algorithm 3, denote* $\mathcal{R}_2^t$ *is a set of RR samples with* $N_t = |\mathcal{R}_t|$, S_2^t *is a candidate solution of phase 2, and*

$$F_u(\mathcal{R}_2^t, S_2^t, \delta) = \frac{\hat{\sigma}(S_2^t)}{1 - \left(1 - \frac{1}{k}\right)^t} + \frac{n}{N_t}\left(\sqrt{c^2 + 2N_t c \frac{\hat{\sigma}(S_2^t)}{\left(1 - \left(1 - \frac{1}{k}\right)^t\right) n}} - c\right)$$

We have $\Pr[\mathsf{OPT} \leq F_u(\mathcal{R}_2^t, S_2^t, \delta)] \geq 1 - \delta$

Proof. Let $\lambda = \sqrt{2c\mu N_t}$, apply inequality (25), we have:

$$\Pr\left[\sum_{i=1}^{N_t} X_g^i(S) - |\mathcal{R}| \cdot \mu \geq -\lambda\right] \leq \exp\left(-\frac{\lambda^2}{2\mu|\mathcal{R}|}\right) \leq \delta \tag{44}$$

Therefore, the following event happens with the probability at least $1 - \delta$:

$$N_t\hat{\mu} - N_t\mu \leq -\sqrt{2c\mu N_t} \Leftrightarrow -N_t(\hat{\mu} - \mu) \geq \sqrt{2c\mu N_t} \tag{45}$$

Solve the above quadratic inequality for μ, we obtain upper-bound for μ is,

$$\mu \leq \max\left\{\hat{\mu}, \hat{\mu} + \frac{1}{N_t}\left(\sqrt{c^2 + 2N_t c\hat{\mu}} - c\right)\right\} \tag{46}$$

$$= \hat{\mu} + \frac{1}{N_t}\left(\sqrt{c^2 + 2N_t c\hat{\mu}} - c\right) \tag{47}$$

Denote $S^0 = \arg\max_{S,|S|\leq k} \hat{\sigma}(S)$, where $\hat{\sigma}$ is calculated over $\mathcal{R}_t^2$. Since the phase of Algorithm 3 selects a candidate solution S_2^t by a greedy strategy. Similar to Theorem 1, we have:

$$\hat{\sigma}(S_2^t) \geq \left(1 - (1 - \frac{1}{k})\right)^t \hat{\sigma}(S^0) \geq \left(1 - (1 - \frac{1}{k})\right)^t \hat{\sigma}(S^*) \tag{48}$$

Replace $\mu = \frac{\sigma(S_2^t)}{n}, \hat{\mu} = \frac{\hat{\sigma}(S_2^t)}{n}$ into (47) and combine it with (48), we have:

$$\mathsf{OPT} = \sigma(S^*) \leq \hat{\sigma}(S^*) + \frac{n}{N_t}\left(\sqrt{c^2 + 2N_t c\frac{\hat{\sigma}(S^*)}{n}} - c\right) \tag{49}$$

$$\leq \frac{\hat{\sigma}(S_2^t)}{\left(1 - (1 - \frac{1}{k})\right)^t} + \frac{n}{N_t}\left(\sqrt{c^2 + 2N_t c\frac{\hat{\sigma}(S_2^t)}{n\left(1 - (1 - \frac{1}{k})\right)^t}} - c\right) \tag{50}$$

which completes the proof. □

Based on above theoretical analysis, the following Theorem Approximation guarantee of IGS algorithm.

Theorem 2. *The Algorithm 3 provides a solution S and an integer t, satisfies:*

- $\Pr[\sigma_U(S) \geq T] \geq 1 - \delta$
- $\Pr[\sigma(S) \geq \left(1 - (1 - \frac{1}{k})^t\right) \mathsf{OPT}] \geq 1 - \delta$

Proof. Since $S = S_1 \cup S_2$ and Lemma 4, we have:

$$\Pr[\sigma_U(S) \geq T] \geq \Pr[\sigma_U(S_1) \geq T] \geq 1 - \delta$$

We consider two following cases:

Case 1: If the algorithm stops with the condition $|\mathcal{R}_2^t| \geq N_{max}$, apply (26) with set S^* and $\mathcal{R}_2$, we have:

$$\Pr[\hat{\sigma}(S^*) \leq (1 - \epsilon_1)\sigma(S^*)] \leq \exp\left\{\frac{-\epsilon_1^2 N_{max}\mathsf{OPT}}{2n}\right\} \tag{51}$$

$$\leq \exp\left\{\frac{-\epsilon_1^2 N_{max} k}{2n}\right\} \quad (\text{Due to } \mathsf{OPT} \geq k) \tag{52}$$

$$\leq \exp\left\{\frac{-\epsilon_1^2 N_{max} t_0}{2n}\right\} \quad (\text{Due to } k \geq t_0) \tag{53}$$

$$\leq \delta_2 / \binom{n}{t_{max}} \leq \delta_2 \tag{54}$$

From (27), we have:

$$\Pr[\sigma(S_2^t) \le \frac{\hat{\sigma}(S_2^t)}{(1+\epsilon_1)}] \le \exp\left\{\frac{-\epsilon_1^2 N_{max}\sigma(S_2^t)}{(2+\frac{2}{3}\epsilon_1)n}\right\} \tag{55}$$

$$\le \exp\left\{\frac{-\epsilon_1^2 N_{max} t}{(2+\frac{2}{3}\epsilon_1)n}\right\} \quad (\text{Due to } \sigma(S_2^t) \ge t) \tag{56}$$

$$\le \delta_2 / \binom{n}{t_{max}} \le \delta_2 \tag{57}$$

Apply an union probability that the events (54) and (57) happen with the probability at most $\delta_1 + \delta_1 = \delta/3$. Assume that they do not happen, we have:

$$\sigma(S_2^t) \ge \frac{\hat{\sigma}(S_2^t)}{1+\epsilon_1} \ge \frac{\left(1-(1-\frac{1}{k})\right)^t \hat{\sigma}(S^0)}{1+\epsilon_1} \tag{58}$$

$$\ge \frac{\left(1-(1-\frac{1}{k})\right)^t \hat{\sigma}(S^*)}{1+\epsilon_1} \tag{59}$$

$$\ge \frac{1-\epsilon_1}{1+\epsilon_1}\left(1-(1-\frac{1}{k})\right)^t \sigma(S^*) \tag{60}$$

$$= \left(\left(1-(1-\frac{1}{k})\right)^t - \frac{2\epsilon_1}{1+\epsilon_1}\left(1-(1-\frac{1}{k})\right)^t\right)\sigma(S^*) \tag{61}$$

$$\ge \left(\left(1-(1-\frac{1}{k})\right)^t - \frac{2\epsilon_1}{1+\epsilon_1}(1-\frac{1}{e})\right)\sigma(S^*) \tag{62}$$

$$\ge \left(\left(1-(1-\frac{1}{k})\right)^t - \epsilon\right)\sigma(S^*) \tag{63}$$

Hence, in this case the algorithm satisfies approximation guarantee with probability at least $1-\frac{\delta}{3}$.

Case 2: If the algorithm stops at any iterator $i, i = 1, 2, \dots, i_{max}$. At this iterator, the condition in line 19 is satisfied, apply Lemma 5 and Lemma 6, the following thing happens with the probability at least $1 - 2i_{max}\delta_2 = 1 - 2\delta/3$:

$$\frac{\sigma(S_2^t)}{\mathsf{OPT}} \ge \frac{F_l(S_2^t, \mathcal{R}_2, \delta_2)}{F_u(S_2^t, \mathcal{R}_2, \delta_2)} \ge \left(1-(1-\frac{1}{k})\right)^t - \epsilon \tag{64}$$

Combine two above cases, the algorithm meets the approximation ratio condition with the probability at least $1 - \delta/3 - 2\delta/3 = 1 - \delta$. □

5. Experiments

In this section, we implement and compare our algorithm IGS to other influence maximization methods about *the influence in general, the influence on priority nodes, running time and memory usage*. The dataset includes several network databases with thousands or even millions nodes and edges (Table 1).

Table 1. Dataset's statistics.

Database	#Nodes	#Edges	Types	Avg. Degree
netHEPT [15]	15 K	59 K	directed	4.1
ENRON [15]	37 K	184 K	directed	5
netPHY [15]	37 K	181 K	directed	13.4
DBLP [15]	655 K	2 M	directed	6.1
TWITTER RETWEET [42]	1 M	2 M	directed	4

5.1. Experimental Settings

All the implementations are on Linux machine with configurations are 2× Intel(R) Xeon(R) CPU E5-2697 v4 @ 2.30GHz and 4 × 16 GB DIMM ECC DDR4 @ 2400MHz.

Algorithm comparisons. Since IMP is an expansion of IM, we compare IGS algorithm with several state-of-the-art IM algorithms including: DSSA [15], BCT [2], OPIM-C [23]. In addition, we use the basic algorithm, Max degree (Degree), which is the common baseline for information diffusion problems. In IMP, there are two factors that impact the solution in practice: the budget (k) of selecting seed node and the priority set of nodes (U). As a result, these two factors also affect the algorithms. From the above observation, we conduct experiments under two settings: varies k and fixed T; varies T and fixed k.

The dataset. For experimental purpose, we choose 5 types of databases from various resources: NetHept, NetPhy, DBLP are citation networks, Email-Enron is communication network [15] and Twitter Retweet is online social networks [42]. The brief of these ones are described on Table 1. These databases are experimented because they are popular in information diffusion problems, especially used in the state-of-the-art algorithms what we are comparing.

Parameter Settings. Graphs are formatted as each edge $e = (u, v) \in E$ has the weight $w(u, v)$ formulated as $w(u, v) = \dfrac{1}{d_{in}(v)}$ where $d_{in}(v)$ is the in-degree of node v [14,15,20].

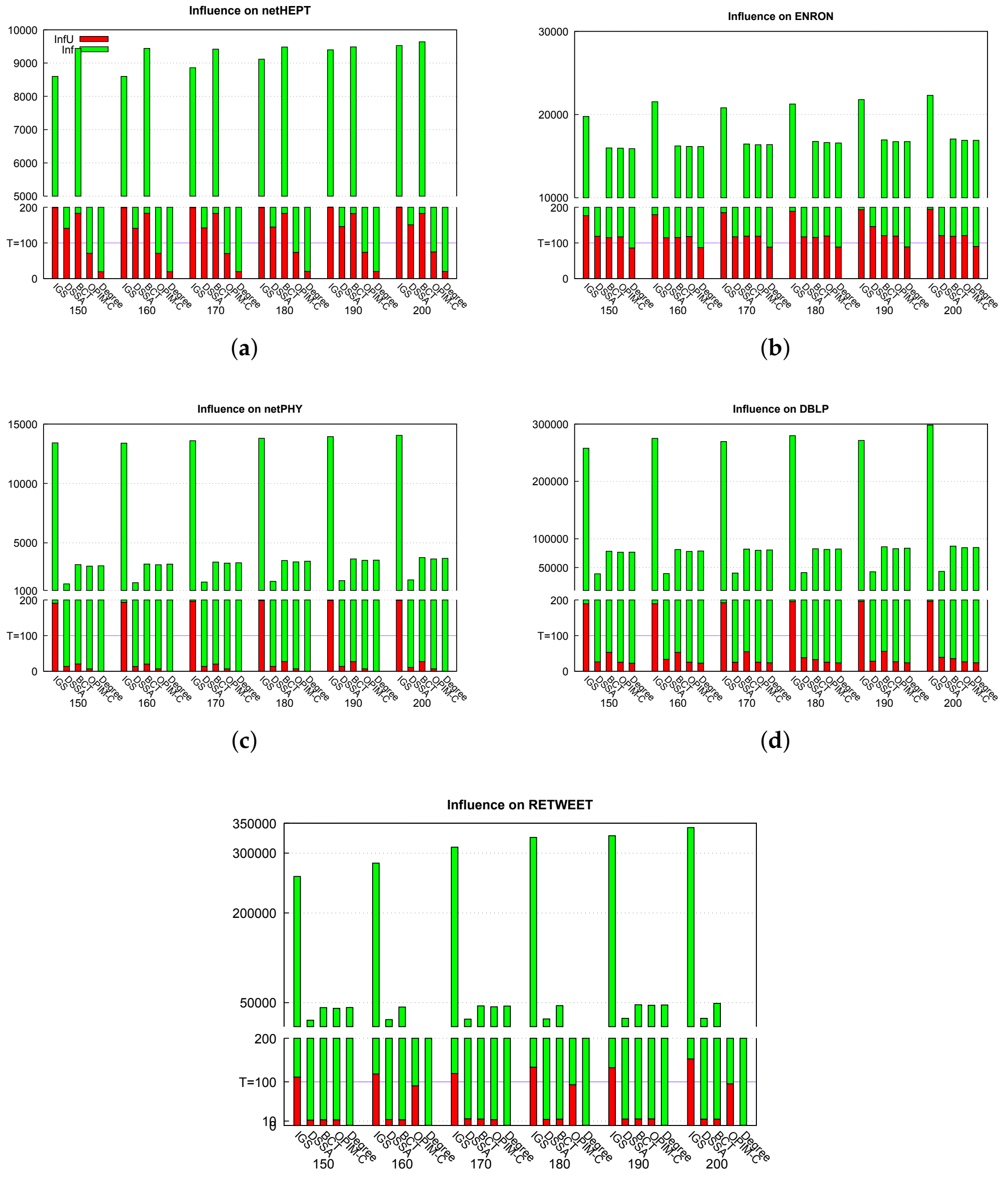
For the first case, k is assigned with 150, 160, 170, 180, 190 and 200, respectively, while T is fixed at 100. In addition, set U is generated with 200 nodes. With the second case, the value of k is fixed at 500. U set includes about 1000 nodes. We change the value of T increasing from 100 to 500. In all experiments, we keep $\epsilon = 0.1$, $\delta = 1/n$ according setting for IM algorithms [14,15,20] and $\alpha = 0.01$.

5.2. Experimental Results

We install IGS to compare with state-of-the-art algorithms such as BCT, DSSA, $\mathsf{OPIM} - C$ and Degree then calculate the spread of influence on all nodes and to U, the priority set, $U \subset V$. Results are shown in following tables and figures.

The Influence. The Figure 2 and the Table 2 indicate IGS outperforms the others when influencing to priority nodes by a given threshold T.

The above figure gives information about the influence values in case k changes from 150 to 200, U includes 200 nodes and the threshold T is 100. The terms "*infU*", "*inf*" mean the influences to set U ($\sigma_U(S)$) and to all nodes ($\sigma(S)$), respectively. These algorithms output differently on various databases. Looking at red bars, we can see IGS approximately affects the set U twice the value of the threshold T on most databases except Re-Tweet but still higher than T . Conversely, the influence on U of the remaining sharply fluctuate according to the databases. While DSSA and BCT influence on U over T with netHEPT and ENRON, they work quite low with the others. $\mathsf{OPIM} - C$ and Degree often affect U much lower than T. Besides, the $\sigma(S)$ of BCT is highest on netHEPT whereas the one of IGS keeps at top in all other cases. In general, the values of $\sigma(S)$ of DSSA, $\mathsf{OPIM} - C$ and Degree have similarities with each others.



(**a**) (**b**)

(**c**) (**d**)

(**e**)

Figure 2. *Comparisons of Influence Spreading* with $k = 100 \to 500$, T = 100 and U size = 200.

Besides, Table 2 describes the experiment while T comes from 100 to 500, k = 500 and enlarge U up to 1K nodes. This setting is to check the case when U is large and when the threshold T is incremental. Certainly, the condition that $k \geq T$ has to be maintained so we fixed k = 500. Looking at bold values, we can see although U and S both become large and T increments gradually, the influence on U of IGS is always significantly higher than T, even up to more than ten times. DSSA, BCT and $\mathsf{OPIM} - C$ also give the outputs over threshold T in many cases, they still have values lower than T = 500 on netPHY, DBLP and RETWEET however. The $\sigma_U(S)$ of Degree is lowest, especially, is only 22.77 on Re-Tweet.

From Figure 2 and Table 2, we can see $\sigma_U(S)$ of IGS is significantly higher than T and produces better results than the state-of-the-art algorithms. This is because IGS always prioritizes affecting U until over the threshold T then affects other nodes as well even with large values of k, U size and T. The other algorithms show that they are not always possible to influence U to exceed the desired threshold. On the whole, the state-of-the-art IM algorithms cannot influence the given priority set as well as IGS can.

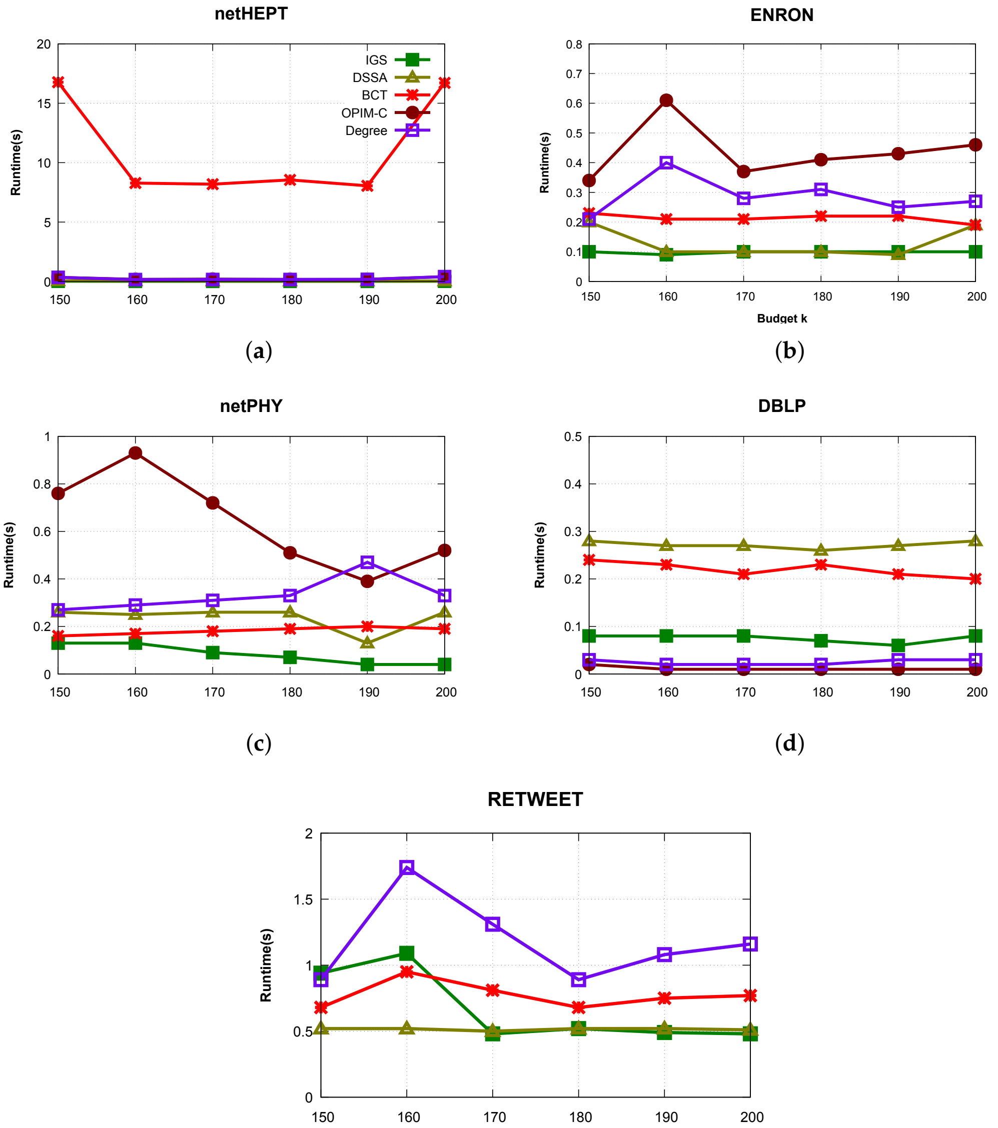
Table 2. Comparisons about $\sigma(S)$ and $\sigma_U(S)$ between IGS and *the others* with k = 500, U size = 1 K and $T = 100 \rightarrow 500$.

			Dataset				
	T		**NetHept**	**Enron**	**netPHY**	**DBLP**	**RETWEET**
IGS	100	$\sigma(S)$	5666.16	14,267.40	1865.92	54,033.50	17,307.70
		$\sigma_U(S)$	**1482.04**	**1075.77**	**1192.84**	**1271.62**	**511.08**
	200	$\sigma(S)$	5581.34	14,162.20	1805.26	53,553.90	18,581.50
		$\sigma_U(S)$	**1478.93**	**1079.74**	**1175.32**	**1267.52**	**491.35**
	300	$\sigma(S)$	5645.40	14,284.80	1773.33	53,240.50	19,459.10
		$\sigma_U(S)$	**1476.08**	**1074.30**	**1153.32**	**1264.79**	**492.39**
	400	$\sigma(S)$	5640.21	14,196.50	1688.53	52,918.80	18,832.20
		$\sigma(S)$	**1468.48**	**1075.68**	**1125.69**	**1260.31**	**490.46**
	500	$\sigma(S)$	5039.45	14,245.50	1593.66	52,130.90	228,801.00
		$\sigma_U(S)$	**1238.54**	**1079.28**	**1104.20**	**1252.70**	**994.40**
DSSA		$\sigma(S)$	4098.63	9960.35	3230.27	58,197.7	38,253.7
		$\sigma_U(S)$	*1093.7*	*857.608*	*174.479*	*474.635*	*168.087*
BCT		$\sigma(S)$	11,088.10	19,901.70	6675.95	117,197.00	77,316.90
		$\sigma_U(S)$	*1280.54*	*1701.60*	*386.49*	*474.635*	*159.77*
OPIM-C		$\sigma(S)$	3779.09	19,326.3	6262.5	112,334	72,026.1
		$\sigma_U(S)$	*600.93*	*894.18*	*194.04*	*459.801*	*173.41*
Degree		$\sigma(S)$	3824.44	19,349.10	6345.86	114,249	73,936
		$\sigma_U(S)$	*292.82*	*779.84*	*164*	*260.94*	*22.77*

Running time. Figure 3 compares running time of these algorithms. They indicate time of IGS gives lowest values on netHEPT, ENRON and netPHY databases. Nevertheless, IGS stays at top 3 on DBLP while it costs highest running time on the remaining of the dataset to find 150 and 160 seed sets but return to top 3 at the other values of budget k. IGS only takes about 0.1 s to find out the seed set in most cases except RETWEET. Besides, the figures also give information about the other algorithms. First, BCT runs significantly slow on netHEPT than the others. This method often stays at top 3 or top 4 on ENRON, DBLP and RETWEET. Second, running time of DSSA and IGS look similiar, while that of OPIM-C and Degree is usually higher than the above two algorithms. As the whole, IGS's running time gives the most stable results and usually runs around the 0.1-s mark.

The time of IGS is fast and stable because of parallel programming and this algorithm costs most of time to find out $S1$ while the loop to calculate $S2$ usually stops at 1–2 rounds. The TRR sampling technique also helps to quickly identify which seeds will affect to the priority U.

Memory Usage. The Table 3 illustrates the memory consumption of IGS and state-of-the-art methods including DSSA, BCT, $\mathsf{OPIM} - C$ and Degree. The smallest numbers are highlighted in bold while the largest ones are in red. The output shows that IGS outperforms the others, especially on small databases with tens of thousands of nodes and from tens to hundreds of thousands of edges such as netHEPT, ENRON, and netPHY. IGS also consumes sharply less memory than $\mathsf{OPIM} - C$ and Degree when testing with larger databases such as DBLP and RETWEET. When IGS spends only more than 130 MB and more than 200 MB, $\mathsf{OPIM} - C$ and Degree spend about four times higher with DBLP and RETWEET, respectively. Besides, DSSA also results less expensive memory usage in all cases. BCT is less stable than IGS and DSSA because it works as DSSA does on ENRON, netPHY, DBLPB and RETWEET but suddenly costs the most memory in NetHEPT.



(**a**) (**b**)

(**c**) (**d**)

(**e**)

Figure 3. Comparisons about *Runtime (s)* with k varies from 150 to 200 between IGS and the others.

TRR sampling technique focuses on finding the seeds that influence the priority U first then Algorithm 3 explores another seeds to push on the seed set. Hence the algorithm 3 saves memory to run loop more than the others because of must not check whether a seed node influences to U set or not. Moreover, the condition of $\frac{F_l(S_2,\mathcal{R}_2,\delta)}{F_u(S_2,\mathcal{R}_2,\delta)} \geq 1-(1-\frac{1}{k})^t - \epsilon$ helps S_2 generated soon without waiting for the stop condition of the repeat.

Finally, our algorithm, IGS, was designed very well to get a balance between the target to influence on the given priority set and the influence that has to propagate to the largest number of nodes. Hence, running time, memory used and the influence of IGS give significantly high results and even more steadily rather than the others in general.

Table 3. *Memory usage (MB)* comparisons between IGS and the others

Dataset	Algorithm	Budget k					
		150	**160**	**170**	**180**	**190**	**200**
NetHEPT	IGS	**9.90**	**9.90**	**9.90**	**9.89**	**9.89**	**9.95**
	DSSA	22.84	22.84	22.84	22.84	22.84	22.84
	BCT	1023.79	1017.52	1021.60	1012.21	1020.18	1020.74
	OPIM-C	47.76	47.91	48.03	48.11	48.30	48.46
	Degree	49.14	49.18	49.48	49.68	49.86	50.13
ENRON	IGS	**16.82**	**16.79**	**16.81**	**16.81**	**16.82**	**16.82**
	DSSA	30.48	28.07	28.07	28.07	28.07	30.48
	BCT	30.35	30.35	30.39	30.39	30.39	30.39
	OPIM-C	27.16	27.20	42.00	27.22	27.25	27.30
	Degree	27.98	28.08	43.77	28.19	28.27	28.41
NetPHY	IGS	**15.18**	**15.18**	**15.18**	**15.18**	**15.18**	**15.04**
	DSSA	52.12	52.12	52.12	52.12	38.50	52.14
	BCT	34.82	34.82	34.82	34.82	34.82	34.80
	OPIM-C	87.88	88.39	88.92	89.31	90.26	90.51
	Degree	92.26	92.71	93.33	93.88	94.68	94.98
DBLP	IGS	**138.66**	**138.66**	**138.66**	**138.66**	**138.66**	**138.66**
	DSSA	152.90	152.87	152.87	152.91	152.91	152.83
	BCT	162.88	162.87	162.87	162.88	162.88	162.89
	OPIM-C	475.05	373.72	373.78	373.95	477.18	477.51
	Degree	500.87	395.00	394.26	395.35	504.52	505.26
RETWEET	IGS	**214.67**	**214.67**	**214.67**	**214.67**	**214.67**	**214.67**
	DSSA	253.14	253.14	253.14	253.14	253.14	253.14
	BCT	282.50	282.50	282.50	282.47	282.50	282.48
	OPIM-C	877.31	874.20	722.91	876.99	886.78	877.80
	Degree	918.53	916.23	756.93	920.00	930.33	921.95

6. Conclusions

In this paper, we investigate the IMP problem, which is a variant of the IM problem with priority constraint that arises in a realistic scenario in which companies or organizations often prioritize influencing potential users during their viral marketing campaigns. The goal of the IMP problem is to select a seed set with k nodes can influence of a given priority set U greater than a threshold T which adjusts the influence of the seed set to the priority set. Although the objective function (influence spread function) is still a monotone and sub-modular function, but when considering the priority constraint the state-of-the-art IM algorithms cannot be applied.

To address this challenge, we propose two algorithms with provable theoretical guarantees, called IG and IGS. We show that IG provides a $\left(1-(1-\frac{1}{k})^t\right)$-approximation solution; IGS is an efficient randomized approximation algorithm based on sampling method that returns a $\left(1-(1-\frac{1}{k})^t-\epsilon\right)$-approximation solution with probability at least $1-\delta$ with $\epsilon > 0, \delta \in (0,1)$ as input parameters of the problem. Experiments on real world social networks show our algorithm outperforms state-of-the-art IM algorithms including DSSA [15], BCT [2] and OPIM [23] in terms of influences, running time, and memory used.

In the future, we are going to improve our algorithm to expand it with large networks to billions scale with acceptable time. In addition, the problem with multiple priority user sets and thresholds is going to be considered.

Author Contributions: methodology and writing–original draft preparation, C.V.P. and D.K.T.H.; investigation Q.C.V., A.N.S.; Conceptualization, H.X.H.; Data curation, Q.C.V. and A.N.S.; Investigation, C.V. P. and D.K.T.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: There is no conflict of interest.

References

1. Kempe, D.; Kleinberg, J.M.; 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. [CrossRef]
2. Nguyen, H.T.; Thai, M.T.; Dinh, T.N. A Billion-Scale Approximation Algorithm for Maximizing Benefit in Viral Marketing. *IEEE/ACM Trans. Netw.* **2017**, *25*, 2419–2429. [CrossRef]
3. Li, Y.; Zhang, D.; Tan, K. Real-time Targeted Influence Maximization for Online Advertisements. *PVLDB* **2015**, *8*, 1070–1081. [CrossRef]
4. Pham, C.V.; Thai, M.T.; Duong, H.V.; Bui, B.Q.; Hoang, H.X. Maximizing misinformation restriction within time and budget constraints. *J. Comb. Optim.* **2018**, *35*, 1202–1240. [CrossRef]
5. Tong, G.A.; Wu, W.; Guo, L.; Li, D.; Liu, C.; Liu, B.; Du, D. An efficient randomized algorithm for rumor blocking in online social networks. In Proceedings of the 2017 IEEE Conference on Computer Communications, INFOCOM 2017, Atlanta, GA, USA, 1–4 May 2017; pp. 1–9. [CrossRef]
6. Budak, C.; Agrawal, D.; El Abbadi, A. Limiting the spread of misinformation in social networks. In Proceedings of the 20th International Conference on World Wide Web, WWW 2011, Hyderabad, India, 28 March–1 April, 2011; pp. 665–674. [CrossRef]
7. Nguyen, H.T.; Cano, A.; Tam, V.; Dinh, T.N. Blocking Self-avoiding Walks Stops Cyber-epidemics: A Scalable GPU-based Approach. *IEEE Trans. Knowl. Data Eng.* **2020**, *32*, 1263–1275. [CrossRef]
8. Nguyen, N.P.; Yan, G.; Thai, M.T. Analysis of misinformation containment in online social networks. *Comput. Netw.* **2013**, *57*, 2133–2146. [CrossRef]
9. Zhang, H.; Alim, M.A.; Li, X.; Thai, M.T.; Nguyen, H.T. Misinformation in Online Social Networks: Detect Them All with a Limited Budget. *ACM Trans. Inf. Syst.* **2016**, *34*, 18:1–18:24. [CrossRef]
10. Zhang, H.; Kuhnle, A.; Zhang, H.; Thai, M.T. Detecting misinformation in online social networks before it is too late. In Proceedings of the 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2016, San Francisco, CA, USA, 18–21 August 2016; pp. 541–548. [CrossRef]
11. Ye, M.; Liu, X.; Lee, W. Exploring social influence for recommendation: A generative model approach. In Proceedings of the 35th International ACM SIGIR conference on research and development in Information Retrieval, SIGIR '12, Portland, OR, USA, 12–16 August 2012; pp. 671–680. [CrossRef]
12. Chen, W.; Collins, A.; Cummings, R.; Ke, T.; Liu, Z.; Rincón, D.; Sun, X.; Wang, Y.; Wei, W.; Yuan, Y. Influence Maximization in Social Networks When Negative Opinions May Emerge and Propagate. In Proceedings of the Eleventh SIAM International Conference on Data Mining, SDM 2011, Mesa, AZ, USA, 28–30 April 2011; pp. 379–390. [CrossRef]
13. Borodin, A.; Filmus, Y.; Oren, J. Threshold Models for Competitive Influence in Social Networks. In Proceedings of the Internet and Network Economics—6th International Workshop, WINE 2010, Stanford, CA, USA, 13–17 December 2010; pp. 539–550. [CrossRef]
14. Tang, Y.; Shi, Y.; Xiao, X. Influence Maximization in Near-Linear Time: A Martingale Approach. In Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, Melbourne, Victoria, Australia, 31 May–4 June 2015; pp. 1539–1554. [CrossRef]
15. 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 Conference 2016, San Francisco, CA, USA, 26 June–1 July 2016; pp. 695–710. [CrossRef]

16. Chen, W.; Yuan, Y.; Zhang, L. Scalable Influence Maximization in Social Networks under the Linear Threshold Model. In Proceedings of the ICDM 2010, The 10th IEEE International Conference on Data Mining, Sydney, Australia, 14–17 December 2010; pp. 88–97. [CrossRef]
17. Chen, S.; Fan, J.; Li, G.; Feng, J.; Tan, K.; Tang, J. Online Topic-Aware Influence Maximization. *PVLDB* **2015**, *8*, 666–677. [CrossRef]
18. Aslay, Ç.; Barbieri, N.; Bonchi, F.; Baeza-Yates, R.A. Online Topic-aware Influence Maximization Queries. In Proceedings of the 17th International Conference on Extending Database Technology, EDBT 2014, Athens, Greece, 24–28 March 2014; pp. 295–306. [CrossRef]
19. Pham, C.V.; Duong, H.V.; Hoang, H.X.; Thai, M.T. Competitive Influence Maximization within Time and Budget Constraints in Online Social Networks: An Algorithmic Approach. *Appl. Sci.* **2019**, *9*, 2274. [CrossRef]
20. 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 2014, Snowbird, UT, USA, 22–27 June 2014; pp. 75–86. [CrossRef]
21. Domingos, P.M.; Richardson, M. Mining the network value of customers. In Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, San Francisco, CA, USA, 26–29 August 2001; pp. 57–66.
22. Leskovec, J.; Krause, A.; Guestrin, C.; Faloutsos, C.; VanBriesen, J.M.; Glance, N.S. 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. [CrossRef]
23. Das, G.; Jermaine, C.M.; Bernstein, P.A.; (Eds.) In Proceedings of the 2018 International Conference on Management of Data, SIGMOD Conference 2018, Houston, TX, USA, 10–15 June 2018. ACM: Rochester, NY, USA, 2018. [CrossRef]
24. Nguyen, H.; Zheng, R. On Budgeted Influence Maximization in Social Networks. *IEEE J. Sel. Areas Commun.* **2013**, *31*, 1084–1094. [CrossRef]
25. Pham, C.V.; Duong, H.V.; Thai, M.T. Importance Sample-Based Approximation Algorithm for Cost-Aware Targeted Viral Marketing. In Proceedings of the Computational Data and Social Networks—8th International Conference, CSoNet 2019, Ho Chi Minh City, Vietnam, 18–20 November 2019; pp. 120–132. [CrossRef]
26. Li, X.; Smith, J.D.; Dinh, T.N.; Thai, M.T. TipTop: (Almost) Exact Solutions for Influence Maximization in Billion-Scale Networks. *IEEE/ACM Trans. Netw.* **2019**, *27*, 649–661. [CrossRef]
27. Barbieri, N.; Bonchi, F.; Manco, G. Topic-aware social influence propagation models. *Knowl. Inf. Syst.* **2013**, *37*, 555–584. [CrossRef]
28. Li, G.; Chen, S.; Feng, J.; Tan, K.-l.; Li, W.-S. Efficient Location-Aware Influence Maximization. In Proceedings of the 34th IEEE International Conference on Data Engineering, ICDE 2018, Paris, France, 16–19 April 2018; pp. 1569–1572. [CrossRef]
29. Wang, X.; Zhang, Y.; Zhang, W.; Lin, X. Efficient Distance-Aware Influence Maximization in Geo-Social Networks. *IEEE Trans. Knowl. Data Eng.* **2017**, *29*, 599–612. [CrossRef]
30. Bharathi, S.; Kempe, D.; Salek, M. Competitive Influence Maximization in Social Networks. In Proceedings of the Internet and Network Economics, Third International Workshop, WINE 2007, San Diego, CA, USA, 12–14 December 2007; pp. 306–311. [CrossRef]
31. Liu, W.; Yue, K.; Wu, H.; Li, J.; Liu, D.; Tang, D. Containment of competitive influence spread in social networks. *Knowl.-Based Syst.* **2016**, *109*, 266–275. [CrossRef]
32. He, X.; Song, G.; Chen, W.; Jiang, Q. Influence Blocking Maximization in Social Networks under the Competitive Linear Threshold Model. In Proceedings of the Twelfth SIAM International Conference on Data Mining, Anaheim, CA, USA, 26–28 April 2012; pp. 463–474. [CrossRef]
33. Lu, W.; Bonchi, F.; Goyal, A.; Lakshmanan, L.V.S. The bang for the buck: Fair competitive viral marketing from the host perspective. In Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2013, Chicago, IL, USA, 11–14 August 2013; pp. 928–936. [CrossRef]
34. Chen, W.; Lakshmanan, L.V.S.; Castillo, C. *Information and Influence Propagation in Social Networks*; Synthesis Lectures on Data Management; Morgan & Claypool Publishers: San Rafael, CA, USA, 2013. [CrossRef]
35. Bozorgi, A.; Samet, S.; Kwisthout, J.; Wareham, T. Community-based influence maximization in social networks under a competitive linear threshold model. *Knowl.-Based Syst.* **2017**, *134*, 149–158. [CrossRef]

36. 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 2019, Macao, China, 10–16 August 2019; pp. 5997–6005. [CrossRef]
37. Farnadi, G.; Babaki, B.; Gendreau, M. A Unifying Framework for Fairness-Aware Influence Maximization. In Proceedings of the Companion of The 2020 Web Conference 2020, Taipei, Taiwan, 20–24 April 2020; pp. 714–722. [CrossRef]
38. Stoica, A.; Han, J.X.; Chaintreau, A. Seeding Network Influence in Biased Networks and the Benefits of Diversity. In Proceedings of the WWW '20: The Web Conference 2020, Taipei, Taiwan, 20–24 April 2020; pp. 2089–2098. [CrossRef]
39. Nguyen, L.N.; Zhou, K.; Thai, M.T. Influence Maximization at Community Level: A New Challenge with Non-submodularity. In Proceedings of the 39th IEEE International Conference on Distributed Computing Systems, ICDCS 2019, Dallas, TX, USA, 7–10 July 2019; pp. 327–337. [CrossRef]
40. Borgs, C.; Brautbar, M.; Chayes, J.T.; Lucier, B. Maximizing Social Influence in Nearly Optimal Time. In Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, OR, USA, 5–7 January 2014; pp. 946–957. [CrossRef]
41. Chung, F.R.K.; Lu, L. Survey: Concentration Inequalities and Martingale Inequalities: A Survey. *Internet Math.* **2006**, *3*, 79–127. [CrossRef]
42. Rossi, R.A.; Ahmed, N.K. *The Network Data Repository with Interactive Graph Analytics and Visualization*; AAAI: Palo Alto, CA, USA, 2015.