Supplementary Influence Maximization Problem in Social Networks

Yapu Zhang, Jianxiong Guo, *Member, IEEE*, Wenguo Yang, and Weili Wu, *Senior Member, IEEE*

***Abstract*— Due to important applications in viral marketing, influence maximization (IM) has become a well-studied problem. It aims at finding a small subset of initial users so that they can deliver information to the largest amount of users through the word-of-mouth effect. The original IM only considers a singleton item. And the majority of extensions ignore the relationships among different items or only consider their competitive interactions. In reality, the diffusion probability of one item will increase when users adopted supplementary products in advance. Motivated by this scenario, we propose a supplementary independent cascade (IC) and discuss the supplementary IM problem. Our problem is NP-hard, and the computation of the objective function is #P-hard. We notice that the diffusion probability will change when considering the impact of its supplementary product. Therefore, the efficient reverse influence sampling (RIS) techniques cannot be applied to our problem directly even though the objective function is submodular. To address this issue, we utilize the sandwich approximation (SA) strategy to obtain a data-dependent approximate solution. Furthermore, we define the supplementary-based reverse reachable (SRR) sets and then propose a heuristic algorithm. Finally, the experimental results on three real datasets support the efficiency and superiority of our methods.**

***Index Terms*— Reverse influence sampling (RIS), sandwich approximation (SA), social networks, supplementary influence maximization (SIM).**

NOMENCLATURE

Notation	*Description*
$G = (V, E)$	Instance of the social network.
n, m	Size of nodes set V and edges set E.
$p_{\mathcal{A}}(u, v)$, $p_{\mathcal{B}}(u, v)$	Diffusion probability on edge (u, v) for product $\mathcal{A}, \mathcal{B}$.
$p^{+}_{\mathcal{B}}(u, v)$	Diffusion probability on edge (u, v) for product $\mathcal{B}$ when v is $\mathcal{A}$-active.
$N^{\text{in}}(v)$, $N^{\text{out}}(v)$	In-neighbors and out-neighbors of v.
$S_{\mathcal{A}}$	Set of seed nodes for product $\mathcal{A}$.
$\hat{S}_{\mathcal{A}}$	St of initial $\mathcal{A}$-active nodes.
$\text{dis}_{g_{\mathcal{A}}}(w, S_{\mathcal{A}})$	Minimum distance from $S_{\mathcal{A}}$ to w in graph $g_{\mathcal{A}}$.

Manuscript received 26 September 2022; revised 20 December 2022 and 30 December 2022; accepted 31 December 2022. This work was supported in part by the National Science Foundation under Grant 1747818 and Grant 1907472 and in part by the National Natural Science Foundation of China under Grant 11991022 and Grant 12071459. *(Corresponding author: Wenguo Yang.)*

Yapu Zhang is with the Institute of Operations Research and Information Engineering, Beijing University of Technology, Beijing 100124, China (e-mail: zhangyapu@bjut.edu.cn).

Jianxiong Guo is with the Advanced Institute of Natural Sciences, Beijing Normal University, Zhuhai 519087, China, and also with the Guangdong Key Laboratory of AI and Multi-Modal Data Processing, Beijing Normal University-Hong Kong Baptist University United International College, Zhuhai 519087, China (e-mail: jianxiongguo@bnu.edu.cn).

Wenguo Yang is with the School of Mathematical Sciences, University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: yangwg@ucas.edu.cn).

Weili Wu is with the Department of Computer Science, University of Texas at Dallas, Richardson, TX 75080 USA (e-mail: weiliwu@utdallas.edu).

Digital Object Identifier 10.1109/TCSS.2023.3234437

I. INTRODUCTION

NOWADAYS, online social media has been integrated into our daily lives. The users become accustomed to receiving and sending information through these media [1], [2]. Therefore, information diffusion in online social networks has been extensively studied by researchers. The influence maximization (IM) problem is to find k users who can deliver information to the largest amount of users through word-of-mouth spread [3]. There are two classic models, namely, independent cascade (IC) and linear threshold (LT) models. Although the IM problem is NP-hard under these two models, Kempe et al. [3] proved that a traditional greedy algorithm could give a $(1 - 1/e)$-approximate solution.

However, the traditional greedy algorithm needs to use Monte-Carlo simulations to estimate the influence spread. This will make the method unable to be applied to large-scale networks. Moreover, the Monte-Carlo simulations cannot give a guarantee when it computes the expected number of influence spread. Some researchers got down to improving the efficiency of the algorithm [4], [5], [6], [7], [8], [9]. Among them, Borgs et al. [6] first proposed an important breakthrough from the view of the reverse influence sampling (RIS). Also, we observe that the IM problem only considers the influence of a single item. Generally, multiple items can diffuse in the same network. There are some multicascade problems proposed [10], [11]. Some of them simply extend the classic IC and LT models, which ignore the relationships among multiple cascades [12], [13]. And most existing works suppose that the entities are pure competitive [14], [15], [16], [17]. That is, each user can only adopt one of them when multiple entities spread in the social network. There are a few diffusion models that allow users to adopt more than one type of entity [18], [19]. However, in their work, they assume that the influence probability of one entity will decrease when users adopted another entity.

In reality, there exists a supplementary relationship among multiple entities. More specifically, the influence probability of one item among users will increase after adopting another item in advance. For instance, one is more inclined to buy AirPods when he or she has already used the iPhone before. Consider the following scenario as an instance. There are two items $\mathcal{A}$ and $\mathcal{B}$ in the social network. The users would adopt item $\mathcal{B}$ with a higher probability if they adopted item $\mathcal{A}$ in advance. Given the distribution of item $\mathcal{A}$, one would like to ask for

a seed of item $\mathcal{B}$ to maximize its expected number of their influence spread.

Motivated by this realistic scenario, we define the supplementary IC (SIC) model and the supplementary IM (SIM) problem. Actually, Lu et al. [20] discussed a similar issue called self IM. Different from their work, our entities in the diffusion model do not be complimentary. More precisely, they assume that item $\mathcal{A}$ can boost the spread of item $\mathcal{B}$, and item $\mathcal{B}$ can also boost the spread of item $\mathcal{A}$. However, our problem does not define that item $\mathcal{B}$ must boost the spread of item $\mathcal{A}$. Meanwhile, in their work, the diffusion probability on each edge is 1. And the probability on each edge is between 0 and 1 in our model. Lin et al. [21] proposed the k-Boosting problem, which asks for k boosted users so that it can maximize influence spread. Boost nodes cannot be active without being influenced by other active nodes but they can be more likely to be active once influenced by other nodes. Unlike our work, this k-Boosting problem focuses on finding initial boosted users instead of seed users.

In summary, our main contributions to this article are as below.

1) Considering the supplementary relationships among items, we extend the IC model and define the SIC model. Based on this diffusion model, the SIM problem is proposed. Given the distribution of supplementary items, the problem is to find the seed users that can maximize their influence spread. We present that this problem is NP-hard, and the computation of the objective function is #P-hard.
2) Fortunately, the objective function is submodular. However, traditional greedy will cause low efficiency. And the original RIS cannot apply to our problem directly since its influence spread will be affected by its supplementary item. To address this problem, we construct its upper bound and lower bound. Using the sandwich approximation (SA) strategy, a data-dependent approximate solution can be obtained.
3) Furthermore, we ignore diffusion probability changes due to the arrival order of the product and its supplementary product. According to this assumption, we design an algorithm based on the supplementary-based reverse reachable (SRR) sets.
4) Finally, we conduct extensive experiments and compare our proposed algorithms with some heuristics in three real-world datasets. These experimental results support the correctness and superiority of our methods.

We organize our article as below. Section II presents some related works. Section III introduces the diffusion model and our problem. We discuss the approximation algorithms with different cases in Section IV. Section V presents the extensive experiments. Finally, we conclude our work in Section VI. For ease of reference, we provide some important notations which are frequently used in nomenclature.

II. Related Work

A. IM Problem

A social network is usually described as a directed graph $G = (V, E)$ with probability $p(u, v)$ on every edge (u, v). The node-set and edge-set represent the users and the relationships among users, respectively. Each node is active or inactive. A node is active if the node adopts one item, or it is inactive otherwise. Formally, given a graph $G = (V, E)$ and an integer k, the IM problem aims to seek k nodes to maximize the final expected number of active nodes. Domingos and Richardson [1] discussed this problem first. They modeled the network as a Markov field and proposed heuristics for maximization. Kempe et al. [3] then considered the IM as combinatorial optimization. They proposed the IC and LT models and showed that it is NP-hard in these two models. Furthermore, they proved that the objective function is nonnegative, monotone nondecreasing, and submodular. Then, there is a $(1 - 1/e)$-approximate solution using the traditional greedy [22]. The IC model is a classic diffusion model, and it works as below. In the beginning, let all seed nodes be active and other nodes be inactive. Next, each newly activated u has one chance to activate its every inactive out-neighbor v with success probability $p(u, v)$. This procedure will terminate if no newly activated nodes are activated. Our model is an extension of this IC model.

B. Reverse Influence Sampling

At each iteration, the greedy needs to estimate the objective function since it is #P-hard to compute the objective function [23], [24]. However, it is time-consuming and lacks a guarantee when using Monte-Carlo simulations to estimate the objective function. The cost-effective lazy forward algorithm [4] and degree discount heuristics [5] have been proposed to improve the efficiency of algorithms. Lately, Borgs et al. [6] first proposed to estimate the objective function using the RIS method. They defined the reverse reachable (RR) set, which contains possible nodes that can reach a selected node. Given a set S, the fraction of RR sets that can be covered by this set S will be used to estimate the objective function for the IM problem. Following this method, some more efficient algorithms are proposed. For instance, Tang et al. [7], [25] devised two-phase influence maximization (TIM), TIM$^+$, and IM via martingales (IMM) algorithms. They proved their algorithms are near-optimal time complexity and can return a $(1 - 1/e - \varepsilon)$-approximate solution with probability at least $1 - \delta$. Here, both ε and δ are parameters in their algorithms. Later, Nguyen et al. [26] devised stop-and-stare algorithm (SSA) and dynamic SSA (DSSA), and Tang et al. [27] proposed the method for online processing of influence maximization (OPIM-C). Although our problem cannot be solved by these techniques directly, our proposed algorithms are still based on the RIS method.

C. Multicascade IM

We observe that the traditional IM problem studied the influence spread of a single item. Some researchers then extend the classic IC and LT models and study some multicascade IM problems [12], [13], [28], [29]. For example, Zhang et al. [12] designed the multiple thresholds model and proposed the profit maximization with multiple adoptions problem. These works ignore the relationships among different items. Furthermore, some works focus on solving the competitive IM problem [14], [15], [16]. Bharathi et al. [14] first discussed the competitive IM and extended the influence diffusion to multiple competing cascades. However, not all relationships among items are

competitive. Lu et al. [20] devised a comparative IC model that studied the interactions from competition to complementarity. Based on this model, they discussed self-IM and complementary IM. According to the complementary relationships, Guo and Wu [10] studied the IM for complementary products. In our work, we define a SIM problem, which considers one item can supplement the diffusion probability of another item.

III. Problem Formulation

We first introduce the diffusion model and problem definition in this section. Then, we discuss some properties of this problem.

A. Diffusion Model

In reality, the owner of one product in one company may prefer to buy another product from this company. Take the Apple carrier as an example. This company manufactures the iPhone and AirPods. It is more likely to buy the AirPods for iPhone users. We say that the iPhone is a supplementary product of the AirPods. Considering this scenario, we study the following model.

Denote the graph $G = (V, E)$ as a social network. Here, V and E represent the users and the relationship between users, respectively. And we assume that $|V| = n$ and $|E| = m$. For each edge (u, v), let $p(u, v) \in [0, 1]$ be the probability of u influences v.

In our work, there are two products. For ease of explanation, we denote these two products as $\mathcal{A}$ and $\mathcal{B}$. Accordingly, for each edge (u, v), the diffusion probability of products $\mathcal{A}$ and $\mathcal{B}$ are $p_{\mathcal{A}}(u, v)$ and $p_{\mathcal{B}}(u, v)$, respectively. Suppose that product $\mathcal{A}$ is the supplementary product of $\mathcal{B}$. That is, the diffusion probability of product $\mathcal{B}$ on each edge (u, v) increases when user v adopted product $\mathcal{A}$ before. We assume that the probability increases to $p^+_{\mathcal{B}}(u, v)$.

Furthermore, for each cascade, let each node be either active (i.e., the adopter of the product) or inactive. We say that a node is $\mathcal{A}$-active (or $\mathcal{B}$-active) when it is activated by $\mathcal{A}$ (or $\mathcal{B}$) cascade. In reality, products $\mathcal{A}$ and $\mathcal{B}$ may not propagate in the network at the same time. Let $\hat{S}_{\mathcal{A}}$ be the set of $\mathcal{A}$-active nodes before the diffusion. Notice that each node in $\hat{S}_{\mathcal{A}}$ cannot influence other nodes to be $\mathcal{A}$-active. Also, products $\mathcal{A}$ and $\mathcal{B}$ may be promoted at a new round of propagation. We denote $S_{\mathcal{A}}$ and $S_{\mathcal{B}}$ as the seed set for products $\mathcal{A}$ and $\mathcal{B}$ for the new round of diffusion process, respectively. Then, we propose the SIC model, an extension of the IC model. Notice that it can be easily extended to the LT model. And the diffusion process is described in discrete time below.

1) In the beginning, each node in $\hat{S}_{\mathcal{A}}$ is $\mathcal{A}$-active, and other nodes are inactive.
2) At time $t = 0$, let each node in $S_{\mathcal{A}}$ and $S_{\mathcal{B}}$ be $\mathcal{A}$-active and $\mathcal{B}$-active, respectively.
3) At time $t > 0$, for $\mathcal{A}$ cascade, each newly activated node u attempts to activate its each inactive out-neighbor v with a success probability $p_{\mathcal{A}}(u, v)$. At the same time, for $\mathcal{B}$ cascade, each newly activated node u also tries to let each inactive out-neighbor v be active. Here, if node v is $\mathcal{A}$-active at time $t'(< t)$, the success probability is $p^+_{\mathcal{B}}(u, v)(> p_{\mathcal{B}}(u, v))$. Otherwise, node v will be $\mathcal{B}$-active with probability $p_{\mathcal{B}}(u, v)$.
4) The process terminates if there are no newly activated nodes for $\mathcal{B}$ cascade.

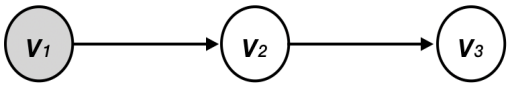

Fig. 1. Example illustrating the diffusion process.

Now, let us see an example for illustrating the diffusion process. As shown in Fig. 1, there are only three nodes v_1, v_2, v_3 and two edges (v_1, v_2) and (v_2, v_3). For each edge $(u, v) \in \{(v_1, v_2), (v_2, v_3)\}$, we suppose that $p_{\mathcal{A}}(u, v) = 1$, $p_{\mathcal{B}}(u, v) = 0.5$ and $p^+_{\mathcal{B}}(u, v) = 0.8$. Let $\hat{S}_{\mathcal{A}} = \{v_3\}$, $S_{\mathcal{A}} = \{v_1\}$, and $S_{\mathcal{B}} = \{v_1\}$. Initially, both v_1 and v_3 are $\mathcal{A}$-active, and v_1 is $\mathcal{B}$-active. Next, node v_2 can be definitely $\mathcal{A}$-active since $p_{\mathcal{A}}(v_1, v_2) = 1$. At the same time, v_1 tries to activate v_2 with 0.5 probability for $\mathcal{B}$ cascade. If v_2 cannot be $\mathcal{B}$-active, then the process terminates. Otherwise, v_2 will be active and attempt to activate v_3 with probability 0.8 next time. Here, the probability is 0.8 since v_3 is $\mathcal{A}$-active before. Finally, the process will end no matter whether v_3 is active or not.

B. Problem Definition

In what follows, we focus on two products diffused in one social network and propose a SIM problem.

We refer to $G = (V, E, P)$ as a directed graph, where $P = (P_{\mathcal{A}}, P_{\mathcal{B}}, P^+_{\mathcal{B}})$. More specifically, $P_{\mathcal{A}}$ and $P_{\mathcal{B}}$ are the influence probabilities of products $\mathcal{A}$ and $\mathcal{B}$, respectively. $P^+_{\mathcal{B}} = \{p^+_{\mathcal{B}}(u, v) \geq p_{\mathcal{B}}(u, v) : (u, v) \in E\}$ means the influence probabilities when v is $\mathcal{A}$-active in advance. Let $\hat{S}_{\mathcal{A}}$ and $S_{\mathcal{A}}$ be the set of initial active nodes and seed nodes for product $\mathcal{A}$, respectively. Given a network G, sets $\hat{S}_{\mathcal{A}}$ and $S_{\mathcal{A}}$, the SIM problem is to seek a seed set $S_{\mathcal{B}} \subseteq V$ with $|S_{\mathcal{B}}| = k$ such that it can maximize the expected number of $\mathcal{B}$-active nodes. Furthermore, we denote by $f(S_{\mathcal{B}})$ the expected number of $\mathcal{B}$-active nodes. Formally, the SIM problem is

$$\max_{S \subseteq V, |S_{\mathcal{B}}| = k} f(S_{\mathcal{B}}) = \sum_{X} \Pr[X] \cdot f_X(S_{\mathcal{B}}) \tag{1}$$

where X is one possible outcome with each edge is deterministic and f_X is the total number of the $\mathcal{B}$-active nodes under the outcome X.

Given a graph $G = (V, E, P)$, an edge (u, v) is $\mathcal{A}$-live (or $\mathcal{B}$-live) if $\mathcal{A}$-active (or $\mathcal{B}$-active) node u can successfully influence node v. Otherwise, this edge is declared to be $\mathcal{A}$-blocked (or $\mathcal{B}$-blocked). To give a better understanding of our problem, we compute the value of the objective function for the example shown in Fig. 1. As aforementioned, we let $p_{\mathcal{A}}(u, v) = 1$, $p_{\mathcal{B}}(u, v) = 0.5$ and $p^+_{\mathcal{B}}(u, v) = 0.8$ for each edge $(u, v) \in \{(v_1, v_2), (v_2, v_3)\}$. Given $\hat{S}_{\mathcal{A}} = \{v_3\}$, $S_{\mathcal{A}} = \{v_1\}$, and $S_{\mathcal{B}} = \{v_1\}$, there are four outcomes as shown in Fig. 2. Notice that the $\mathcal{B}$-live and $\mathcal{B}$-blocked edges are denoted by solid arrows and solid arrows with crosses,

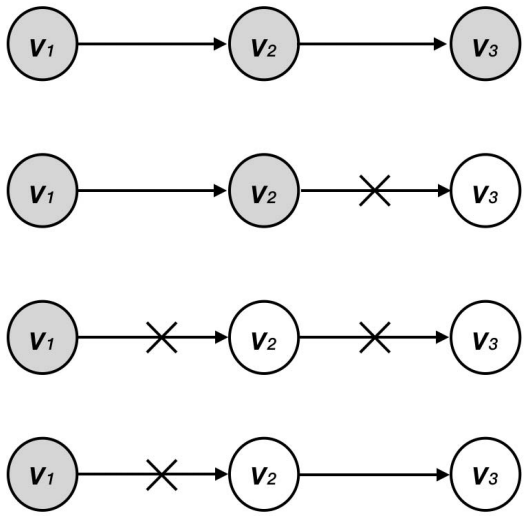

Fig. 2. Example illustrating the computation of the objective function.

respectively. Here, (v_1, v_2) is $\mathcal{B}$-live (resp. $\mathcal{B}$-block) with probability 0.5 (resp. 0.5). Since $v_3 \in \hat{S}_{\mathcal{A}}$, (v_2, v_3) is $\mathcal{B}$-live (resp. $\mathcal{B}$-block) with probability 0.8 (resp. 0.2). In the first case, edges (v_1, v_2) and (v_2, v_3) are $\mathcal{B}$-live, that is, three nodes can be $\mathcal{B}$-active with probability 0.4. In the second case, (v_1, v_2) is $\mathcal{B}$-live and (v_2, v_3) is $\mathcal{B}$-block, that is, two nodes v_1 and v_2 are $\mathcal{B}$-active with probability 0.1. The last two cases will cause only one node to be $\mathcal{B}$-active and the total probability is 0.5. Then, we have $f(S_{\mathcal{B}}) = 3 \cdot 0.4 + 2 \cdot 0.1 + 1 \cdot 0.5 = 1.9$.

C. Property

Theorem 1: The SIM problem is NP-hard, and it is #P-hard to compute its objective function.

Proof: The SIM problem is totally equivalent to the IM problem under the classic IC model when $\hat{S}_{\mathcal{A}} = \emptyset$ and $S_{\mathcal{A}} = \emptyset$. Notice that the IM problem is NP-hard [3]. Thus, the SIM problem is NP-hard. By a similar argument, for any given set $S_{\mathcal{B}}$, it is #P-hard to compute its objective function [23], [24]. □

Theorem 2: The objective function of the SIM problem is nonnegative, nondecreasing monotone, and submodular.

Proof: By definition, it suffices to prove the submodularity of the objective function. Denote by X a deterministic outcome. Like the claim in [3], for a given seed set $S_{\mathcal{B}}$, a node v can be $\mathcal{B}$-active if and only if there is a $\mathcal{B}$-live path from some node in $S_{\mathcal{B}}$ to v in this outcome X. Notice that we call a path $\mathcal{B}$-live if each edge in this path is $\mathcal{B}$-live. Let $R(v, X)$ be the set of nodes that can be $\mathcal{B}$-active after the influence spread of v in this outcome X. Then, we have $f_X(S_{\mathcal{B}}) = \bigcup_{v \in V} R(v, X)$. Given any two sets $S \subseteq T$ and a node $v \in V \setminus T$, it suffices to prove

$$f_X(T \cup \{v\}) - f_X(T) \leq f_X(S \cup \{v\}) - f_X(S) \tag{2}$$

$f_X(S \cup \{v\}) - f_X(S)$ is the number of all nodes that are included in $R(v, X)$ but not included in $\bigcup_{u \in S} R(u, X)$. Then, (2) holds since $S \subseteq T$. Furthermore, we have $f(S_{\mathcal{B}}) = \sum_X \Pr[X] \cdot f_X(S_{\mathcal{B}})$ and the theorem follows. □

Algorithm 1 Max-Coverage

Input: $\mathcal{R}, k$
Output: $S_{\mathcal{B}}$
1: Initialize $S_{\mathcal{B}} = \emptyset$
2: **for** $i = 1$ *to* k **do**
3: $v \leftarrow$ the node that covers the most RR-sets in $\mathcal{R}$
4: $S_{\mathcal{B}} \leftarrow S_{\mathcal{B}} \cup \{v\}$
5: Remove all RR-sets in which v appears
6: **end for**

IV. Approximation Algorithms

Next, we discuss how to tackle the SIM problem. To address it, we first consider a special case when $S_{\mathcal{A}} = \emptyset$ and then study the case when $\hat{S}_{\mathcal{A}} = \emptyset$. We show that the methods when $\hat{S}_{\mathcal{A}} = \emptyset$ can easily be used for the general SIM problem.

A. Case When $S_{\mathcal{A}} = \emptyset$

According to the definition, $\mathcal{A}$-active nodes influenced by seed set $S_{\mathcal{A}}$ is underspecified. In the meantime, the spread of seed set $S_{\mathcal{B}}$ at each step is related to the spread of $S_{\mathcal{A}}$. Thus, the SIM problem is not order-independent. It poses challenges to solving this problem.

First, we discuss a special case of the SIM when $S_{\mathcal{A}} = \emptyset$. In this case, we can determine the state of each edge in advance. A simple way to estimate the influence spread using Monte-Carlo methods. More specifically, we first sample a number of deterministic graphs. A deterministic graph is generated as follows. Given a graph $G = (V, E, P)$ and set $\hat{S}_{\mathcal{A}}$, we construct a new propagation probability $Q = \{q(u, v) : (u, v) \in E\}$, where

$$q(u, v) = \begin{cases} p^+_{\mathcal{B}}(u, v), & \text{if } v \in \hat{S}_{\mathcal{A}} \\ p_{\mathcal{B}}(u, v), & \text{otherwise.} \end{cases}$$

Denote by $\Omega = (V, E, Q)$ a graph with edge probability distribution Q. For a deterministic graph $g \sim \Omega$, each edge (u, v) is live with probability $q(u, v)$ and blocked with probability $1 - q(u, v)$. Then, for a deterministic graph g, the influence spread $f_g(S_{\mathcal{B}})$ is the number of nodes that $S_{\mathcal{B}}$ can reach. Given a set of deterministic graphs $\mathcal{G}$, $f(S_{\mathcal{B}})$ can be approximated by $\sum_{g \in \mathcal{G}} (f_g(S_{\mathcal{B}})/|\mathcal{G}|)$. Obviously, this method cannot provide a theoretical guarantee and it is hard to handle large-scale social networks.

In this article, we use the RIS [7] technique to estimate our objective function. This idea is motivated by the RR set (RR-set). A random RR-set R is generated under the graph $\Omega = (V, E, Q)$ as follows: 1) sampling a deterministic graph $g \sim \Omega$; (2) selecting a node v randomly; and 3) collecting all nodes in g that can reach to v into R.

Lemma 1 [25]: Let $\mathcal{R}$ be a collection of RR-sets for Ω. Given a seed set $S_{\mathcal{B}}$, we say that $S_{\mathcal{B}}$ covers $R \in \mathcal{R}$ if $S_{\mathcal{B}} \cap R \neq \emptyset$. Then, $f(S_{\mathcal{B}}) = n \cdot \mathbb{E}[S_{\mathcal{B}} \text{ covers } R]$, where $\mathbb{E}[\cdot]$ is an expected operator.

According to the above lemma, we know that our problem can be solved by the maximum coverage problem [30]. This problem asks for k nodes to cover the largest size of the

given sets. As shown in Algorithm 1, the greedy algorithm can return a $(1-1/e)$-approximate solution for the maximum coverage problem. Given a set of RR-sets $\mathcal{R}$, we repeatedly choose the node v that can cover the most RR-sets in $\mathcal{R}$. At each iteration, we delete from $\mathcal{R}$ all RR-sets covered by v. The procedure stops until k nodes are selected. The time complexity of Algorithm 1 is $O(k\sum_{R\in\mathcal{R}}|R|)$.

Given a set $S_{\mathcal{B}}$, let $\mathbb{I}_{S_{\mathcal{B}}}$ be an indicator function. That is, $\mathbb{I}_{S_{\mathcal{B}}}(R)=1$ if $S_{\mathcal{B}}\cap R\neq\emptyset$, or $\mathbb{I}_{S_{\mathcal{B}}}(R)=0$ otherwise. Then, we have $f(S_{\mathcal{B}})$ can be estimated by $(1/|\mathcal{R}|)\cdot\sum_{R\in\mathcal{R}}\mathbb{I}_{S_{\mathcal{B}}}(R)$ for a given set of RR-sets $\mathcal{R}$. The estimation is much more closely when the size of RR-sets is sufficiently large.

Lemma 2 [25]: Let OPT be the optimal value of the SIM problem when $S_{\mathcal{A}}=\emptyset$. The max-coverage algorithm can return a $(1-1/e-\varepsilon)$-approximate solution with at least probability $1-\delta$, if

$$|\mathcal{R}|\geq\frac{2n\cdot\left((1-1/e)\sqrt{\ln\frac{2}{\delta}}+\sqrt{(1-1/e)\left(\ln\binom{n}{k}\ln\frac{2}{\delta}\right)}\right)^2}{\varepsilon^2\mathrm{OPT}}. \tag{3}$$

Furthermore, there are some methods to estimate the size of $\mathcal{R}$. For instance, Tang et al. proposed TIM, TIM$^+$ [7], and IMM algorithms [25]. Also, some efficient algorithms, such as SSA, DSSA [26], and OPIM-C [27], are proposed. We can use these methods to complete our algorithm. Actually, if we use k instead of OPT in (3), it can provide an upper bound of the size of RR-sets since $|S_{\mathcal{B}}|=k$.

B. Case When $\hat{S}_{\mathcal{A}}=\emptyset$

We propose two methods to address this SIM problem. One approximate algorithm utilizes the SA strategy. The other one is a heuristic algorithm using the RIS technique with the SRR sets.

1) Sandwich Approximation: As shown in Theorem 2, the objective function of the SIM problem is submodular. A simple way is to iteratively choose the node with the maximum marginal gain until the size of the selected node is k. This method can return a $(1-1/e)$-approximate solution [22]. However, it is hard to compute the marginal gain $\Delta_v f(S_{\mathcal{B}})=f(S_{\mathcal{B}}\cup\{v\})-f(S_{\mathcal{B}})$.

Algorithm 2 gives the process to compute $f(S_{\mathcal{B}})$ by the Monte-Carlo method with simulation number r. Let $N^{\text{in}}(u)=\{v|(v,u)\in E\}$ and $N^{\text{out}}(u)=\{v|(u,v)\in E\}$. At each iteration, we first generate a realization $g_{\mathcal{A}}\sim\Omega=(G,V,P_{\mathcal{A}})$ and a queue $Q=\{S_{\mathcal{B}}\}$. Then, we simulate a possible size of $\mathcal{B}$-active nodes using breadth-first search (BFS). Notice that the propagation probability changes from $p_{\mathcal{B}}(u,w)$ to $p^+_{\mathcal{B}}(u,w)$ when w is $\mathcal{A}$-active in advance. Let $\mathrm{dis}(w,S_{\mathcal{B}})$ be the minimum distance from $S_{\mathcal{B}}$ to w for a simulation of cascade $\mathcal{B}$. In addition, $\mathrm{dis}_{g_{\mathcal{A}}}(w,S_{\mathcal{A}})$ means the minimum distance from $S_{\mathcal{A}}$ to w in a deterministic graph $g_{\mathcal{A}}$. For a random number $\alpha\in[0,1]$, w can be influenced by u if $\alpha\leq p_{\mathcal{B}}(u,w)$. Also, w can be influenced by u if $\mathrm{dis}(w,S_{\mathcal{B}})>\mathrm{dis}_{g_{\mathcal{A}}}(w,S_{\mathcal{A}})$ and $\alpha\leq p^+_{\mathcal{B}}(u,w)$.

The time complexity of BFS is $O(k(n+m))$ and the total running time of computing $f(S_{\mathcal{B}})$ is $O(k(n+m)r)$. Using the Monte-Carlo method, we should compute the marginal gain of each node at each iteration. Obviously, it is time-consuming. Therefore, we should find an efficient algorithm. Since the uncertainty of probability on each edge for product $\mathcal{B}$, we cannot estimate the objective function using the RIS technique.

Algorithm 2 Estimation of Objective Function

Input: $G=(V,E,P)$, $S_{\mathcal{A}}$, $S_{\mathcal{B}}$ and r
Output: $f(S_{\mathcal{B}})$

Initialize $Total\leftarrow 0$
for $i=1$ to r **do**
  Initialize $f(S_{\mathcal{B}})\leftarrow 0$
  Generate a realization $g_{\mathcal{A}}\sim\Omega=(G,V,P_{\mathcal{A}})$
  Initialize a queue $Q\leftarrow S_{\mathcal{B}}$
  Initialize $dis(v,S_{\mathcal{B}})\leftarrow 0$ for $v\in S_{\mathcal{B}}$ and $dis(u,S_{\mathcal{B}})\leftarrow\infty$ for $u\in V\setminus S_{\mathcal{B}}$
  Mark each node in $S_{\mathcal{B}}$ as visited
  while Q is not empty **do**
    $u\leftarrow Q.dequeue()$
    for each non-visited $w\in N^{out}(u)$ **do**
      α is selected from $[0,1]$ uniformly at random
      if $\alpha\leq p_{\mathcal{B}}(u,w)$ or $(dis(u,S_{\mathcal{B}})+1>dis_{g_{\mathcal{A}}}(w,S_{\mathcal{A}})$ and $p\leq p^+_{\mathcal{B}}(u,w))$ **then**
        $dis(w,S_{\mathcal{B}})\leftarrow dis(u,S_{\mathcal{B}})+1$
        $Q.enqueue(w)$ and mark w as visited
        $f(S_{\mathcal{B}})\leftarrow f(S_{\mathcal{B}})+1$
      end if
    end for
  end while
  $Total\leftarrow Total+f(S_{\mathcal{B}})$
end for
return $f(S_{\mathcal{B}})\leftarrow Total/r$

To tackle the SIM, we devise a way based on the SA strategy [20]. This method can provide a data-dependent approximate solution according to its submodular lower and upper bounds. Next, we get down to constructing the upper and lower bounds of the objective function as follows. Given $G=(V,E,P)$ and $S_{\mathcal{A}}$, let $V_{\mathcal{A}}$ be a set of nodes that contains all the nodes that $S_{\mathcal{A}}$ can reach. Furthermore, we construct a new propagation probability $Q^+=\{q^+(u,v):(u,v)\in E\}$, where

$$q^+(u,v)=\begin{cases}p^+_{\mathcal{B}}(u,v), & \text{if } v\in V_{\mathcal{A}}\\ p_{\mathcal{B}}(u,v), & \text{otherwise.}\end{cases}$$

Let R^+ be a random RR-set under graph $\Omega^+=(V,E,Q^+)$. Then, $f^+(S_{\mathcal{B}})=n\cdot\mathbb{E}[S_{\mathcal{B}}\text{ covers }R^+]$ is the submdoular upper bound of the objective function. Similarly, let R^- be a random RR-set under $\Omega^-=(V,E,P_{\mathcal{B}})$. We have $f^-(S_{\mathcal{B}})=n\cdot\mathbb{E}[S_{\mathcal{B}}\text{ covers }R^-]$ is the submdoular lowers bound.

In Algorithm 3, an approximate solution to the upper bound is obtained as follows. We estimate the lower bound of OPT for f^+ using the method in [25]. Then, we create a collection of RR-sets $\mathcal{R}^+$, where $|\mathcal{R}^+|$ is computed by (3) with its lower bound replacing OPT. According to Algorithm 1, an approximate solution S_u is obtained. Similarly, we have an approximate solution S_l to the lower bound f^-. The time complexity

Algorithm 3 SA

Input: $G = (V, E, P), k, S_{\mathcal{A}}$
Output: $S_{\mathcal{B}}$

Initialize $S_{\mathcal{B}} \leftarrow \emptyset$
$S_l \leftarrow$ the approximate solution to the upper bound
$S_o \leftarrow$ a solution to the original problem
$S_u \leftarrow$ the approximate solution to the lower bound
$S_{\mathcal{B}} \leftarrow \arg\max_{S \in \{S_l, S_o, S_u\}} f(S)$

of calculating the solutions S_l and S_u is $O(k \sum_{R \in \mathcal{R}^-} |R|)$ and $O(k \sum_{R \in \mathcal{R}^+} |R|)$, respectively. Let S_o be a solution to f with any method. Here, we estimate $f(v)$ for each node $v \in V$ and select k nodes with the maximum value of f as our original solution. SA is to select the best solution to the objective function f among these three solutions. That is, we select $S_{\mathcal{B}} = \arg\max_{S \in \{S_l, S_o, S_u\}} f(S)$ as our final result. The time complexity of selecting the best solution $S_{\mathcal{B}}$ is $O(k(n+m)r)$.

Theorem 3: Let S^* be the optimal solution for the original problem. At least $1 - 2\delta$ probability, Algorithm 3 can derive a

$$\max\left\{\frac{f(S_u)}{f^+(S_u)}, \frac{f^-(S^*)}{f(S^*)}\right\} \cdot (1 - 1/e - \varepsilon) \tag{4}$$

approximate solution.

Proof: Let S_u^* and S_l^* be the optimal solutions for maximizing the lower bound and upper bound

$$\begin{aligned} f(S_u) &= \frac{f(S_u)}{f^+(S_u)} \cdot f^+(S_u) \\ &\geq \frac{f(S_u)}{f^+(S_u)} \cdot (1 - 1/e - \varepsilon) \cdot f^+\left(S_u^*\right) \\ &\geq \frac{f(S_u)}{f^+(S_u)} \cdot (1 - 1/e - \varepsilon) \cdot f^+\left(S^*\right) \\ &\geq \frac{f(S_u)}{f^+(S_u)} \cdot (1 - 1/e - \varepsilon) \cdot f\left(S^*\right) \end{aligned}$$

and

$$\begin{aligned} f(S_l) &\geq f^-(S_l) \geq (1 - 1/e - \varepsilon) \cdot f^-(S_l^*) \\ &\geq (1 - 1/e - \varepsilon) \cdot f^-(S^*) \\ &\geq \frac{f^-(S^*)}{f(S^*)}(1 - 1/e - \varepsilon) \cdot f(S^*). \end{aligned}$$

Since Algorithm 3 returns a solution $S_{\mathcal{B}} = \arg\max_{S \in \{S_l, S_o, S_u\}} f(S)$, we have

$$f(S_{\mathcal{B}}) \geq \max\left\{\frac{f(S_u)}{f^+(S_u)}, \frac{f^-(S^*)}{f(S^*)}\right\} \cdot (1 - 1/e - \varepsilon) f(\mathrm{OPT}).$$

□

2) Heuristic Algorithm: As mentioned before, we cannot use the RIS technique to estimate the objective function since the uncertainty of probability on each edge for product $\mathcal{B}$. Furthermore, we consider fixing the diffusion probability for $\mathcal{B}$ cascade.

We suppose that the diffusion probability is $p^+_{\mathcal{A}}(u, v)$ if and only if v can be $\mathcal{A}$-active in the diffusion process. That is, we still think the probability on edge (u, v) is $p^+_{\mathcal{A}}(u, v)$ even if v is $\mathcal{A}$-active after being $\mathcal{B}$-active. Accordingly, we define a random SRR set and compute it using Algorithm 4. Similar to

Algorithm 4 SRR Set

Input: $G = (V, E, P), k$ and $S_{\mathcal{A}}$
Output: T

v is a random node from V
Initialize a set $T \leftarrow \{v\}$
Initialize a queue $Q \leftarrow S_{\mathcal{A}}$
Mark all nodes in $S_{\mathcal{A}}$ as $\mathcal{A}$-active
while Q is not empty **do**
  $u \leftarrow Q.dequeue()$
  for each $\mathcal{A}$-inactive node $w \in N^{out}(u)$ **do**
    $\alpha_{\mathcal{A}}$ is selected from [0, 1] uniformly at random
    if $\alpha_{\mathcal{A}} \leq p_{\mathcal{A}}(u, w)$ **then**
      $Q.enqueue(w)$ and mark w as $\mathcal{A}$-active
    end if
  end for
end while
Clear Q and let $Q \leftarrow \{v\}$
Mark v as visited
while Q is not empty **do**
  $u \leftarrow Q.dequeue()$
  for each non-visited node $w \in N^{in}(u)$ **do**
    $\alpha_{\mathcal{B}}$ is selected from [0, 1] uniformly at random
    if $\alpha_{\mathcal{B}} \leq p_{\mathcal{B}}(w, u)$ or (u is $\mathcal{A}$-active and $\alpha_{\mathcal{B}} \leq p^+_{\mathcal{B}}(w, u)$) **then**
      $Q.enqueue(w)$ and mark w as visited
      $T \leftarrow T \cup \{w\}$
    end if
  end for
end while

the RR set, a random SRR set contains the reachable nodes from a randomly selected node with this assumption.

To obtain a random SRR set, we first determine an outcome for $\mathcal{A}$ cascade. Given seed set $S_{\mathcal{A}}$ and a random node v, Algorithm 4 first finds the nodes can be $\mathcal{A}$-active. More specifically, it utilizes the forward BFS to find the nodes that can be reached from node $S_{\mathcal{A}}$. Then, Algorithm 4 aims to find that the nodes can reach v based on the outcome for $\mathcal{A}$ cascade. If u is $\mathcal{A}$-active, w can be added into the SRR set T with probability $p^+(w, u)$. If u is not $\mathcal{A}$-active, w can be added into the SRR set T with probability $p(w, u)$. Here, we generate the SRR set T using the backward BFS.

Furthermore, we conclude the following lemma.

Lemma 3: Let T be a random SRR set. $f_1^+(S_{\mathcal{B}}) = n \cdot \mathbb{E}[S_{\mathcal{B}} \text{ covers } T]$ is a submodular upper bound of the objective function for any set $S_{\mathcal{B}}$.

Proof: By definitions, $f(S_{\mathcal{B}})/n$ is the probability that a selected node v is $\mathcal{B}$-active by $S_{\mathcal{B}}$. That is, there is a reachable path from one node in $S_{\mathcal{B}}$ to node v. Notice that $S_{\mathcal{B}}$ covers T means there is a reachable path from a node in $S_{\mathcal{B}}$ to node v without the order for cascades. Thus, f_1^+ is an upper bound of f.

Moreover, f_1^+ is submodular. Given sets $S_1 \subseteq S_2 \subseteq V$ and node $v \in V \setminus S_2$, $(f^+(S_1 \cup \{v\}) - f^+(S_1)/n)$ is the probability that v can cover T but S_1 cannot do it. Therefore,

TABLE I
STATISTICS OF DATASETS

Name	#Nodes	#Edges	Type	Avg. Deg
NetScience	1.589K	2.742K	Undirected	3.45
HepTh	27.7K	352.8K	Directed	12.7
Stanford	281.9K	2.3M	Directed	8.2

$(f^+(S_1 \cup \{v\}) - f^+(S_1)/n) \geq (f^+(S_2 \cup \{v\}) - f^+(S_2)/n)$ and the lemma follows. □

Using the RIS technique, we can estimate $n \cdot \mathbb{E}[S_{\mathcal{B}} \text{ covers } T]$ and obtain an approximate solution for maximizing $n \cdot \mathbb{E}[S_{\mathcal{B}} \text{ covers } T]$. We will regard this solution as a solution of f.

C. Case When $S_{\mathcal{A}} \neq \emptyset$ and $\hat{S}_{\mathcal{A}} \neq \emptyset$

Furthermore, we can use the algorithms when $\hat{S}_{\mathcal{A}} = \emptyset$ to solve the case when $S_{\mathcal{A}} \neq \emptyset$ and $\hat{S}_{\mathcal{A}} \neq \emptyset$. More specifically, we use graph $\hat{G} = (V, E, \hat{P})$ as the initial graph $G = (V, E, P)$, where $\hat{P}_{\mathcal{A}} = P_{\mathcal{A}}$; $\hat{P}_{\mathcal{B}} = \{\hat{p}_{\mathcal{B}}(u, v) : (u, v) \in E\}$ with $\hat{p}_{\mathcal{B}}(u, v) = p^+_{\mathcal{B}}(u, v)$ if $v \in \hat{S}_{\mathcal{A}}$ and $\hat{p}_{\mathcal{B}}(u, v) = p_{\mathcal{B}}(u, v)$ otherwise; $\hat{P}^+_{\mathcal{B}} = P^+_{\mathcal{B}}$.

Notice that f^- will be 0 when $p_{\mathcal{B}}(u, v) = 0$ for each edge (u, v). Here, $p_{\mathcal{B}}(u, v) = 0$ means that the node cannot be $\mathcal{B}$-active if it is not $\mathcal{A}$-active. We can only use its upper bound to obtain a $(f(S_u)/f^+(S_u)) \cdot (1 - 1/e - \varepsilon)$-approximate solution.

V. EXPERIMENTS

In the following, we conduct several experiments using our proposed algorithms and other heuristic methods. By comparison, these experiments support the effectiveness and efficiency of our methods.

A. Experimental Settings

1) Datasets: We do our experiments based on three real networks: 1) Netscience [31] captures a co-authorship among scientists working on network theory and experiment; 2) HepTh [32] is a citation network from the e-print arXiv. If paper i cites paper j, there is an edge from i to j; and 3) Stanford [32] is generated from the Stanford University website. Each node means pages, and each edge means hyperlinks between them. For the undirected graph, we use two reversed directed edges to present each undirected edge. And the statistics of these networks is shown in Table I.

2) Influence Probability: There are three influence probabilities in our problem. More specifically, $P_{\mathcal{A}}$ and $P_{\mathcal{B}}$ are the influence probabilities of products $\mathcal{A}$ and $\mathcal{B}$, respectively. And $P^+_{\mathcal{B}}$ is the influence probability of $\mathcal{B}$ when considering the impact of its supplementary product $\mathcal{A}$. For $P_{\mathcal{A}}$ and $P_{\mathcal{B}}$, we sample each element $p_{\mathcal{A}}(u, v)$ and $p_{\mathcal{B}}(u, v)$ from $[0, 0.1]$ uniformly. According to the method in [21], we let $p^+_{\mathcal{B}}(u, v) = 1 - (1 - p_{\mathcal{B}}(u, v))^{\beta}$, where $\beta > 1$ is the supplementary parameter. Unless otherwise specified, we set $\beta = 2$.

3) Selection of Seeds: We should fix the seed set $S_{\mathcal{A}}$ before the diffusion process, and we use two methods to give set $S_{\mathcal{A}}$ as follows.

1) We select 20 influential nodes using IMM algorithm [25]. Generally, a company would like to choose influential persons as their initial users to promote their products.
2) We choose 200 nodes randomly. In reality, some users will adopt a product spontaneously. For ease of reference, we refer to the above two cases as case-1 and case-2, respectively.

4) Algorithms: The influence probability of each edge is fixed when applying IM algorithms. However, the probability can change for cascade $\mathcal{B}$ in our problem. To the best of our knowledge, there is no existing algorithm that can apply to the SIM problem and we mainly consider several algorithms as listed below.

1) *SA:* This method is presented in Algorithm 3. We refer to IMM [25] to solve the upper and lower bounds.
2) *SRR:* This algorithm is a max-coverage with SRR sets generated by Algorithm 4.
3) *Random:* It is to select k nodes randomly, considered a baseline.
4) *OutDegree:* This strategy is to choose k nodes with the maximum out-degree.
5) *PageRank:* The k nodes with the largest PageRank scores as the solution [33]. And we let the error value and damping coefficient be 10^{-6} and 0.85, respectively.
6) *A linear algorithm for influence maximization (LAIM):* This is a linear time algorithm for IM in large-scale social networks [8]. Here, we set the parameter $\gamma = 4$ and let the influence probability for cascade $\mathcal{B}$ be $p_{\mathcal{B}}(u, v)$ for each edge (u, v) without considering the influence of cascade $\mathcal{A}$.

5) Settings: For both the SA and SRR algorithms, we fix $\varepsilon = 0.5$ and $\delta = 1/|V|$. To ensure the fairness of the experiments, we use 10 000 Monte-Carlo simulations to estimate our objective function. All experiments run on a machine with a 3.6 GHz, quad-core processor, and 8 GB memory.

B. Experimental Results

1) Performance With Different Budget k: First, for different budgets k, we evaluate the influence spread of different seed sets $S_{\mathcal{B}}$ obtained from different algorithms. As shown in Figs. 3 and 4, our proposed algorithms (i.e., SA and SRR) outperform other algorithms for both case-1 and case-2. And our algorithms perform particularly well in HepTh and Stanford networks. This is because the difference in objective function values is more obvious in large networks. Meanwhile, the results of case-1 and case-2 are similar. The influence spread increases when the budget k increases in three networks. And the results of SA and SRR are very close. These imply the efficiency of our methods.

2) Approximation Ratio of the SA Strategy: The SA algorithm can derive a data-dependent approximation ratio. To show this approximation ratio, we present the results for the upper bound, original function, and lower bound with the set $S_{\mathcal{B}}$ returned by the SA algorithm. Likewise, we conduct our experiments in two cases. As shown in Figs. 5 and 6, the results among these three functions are very close no

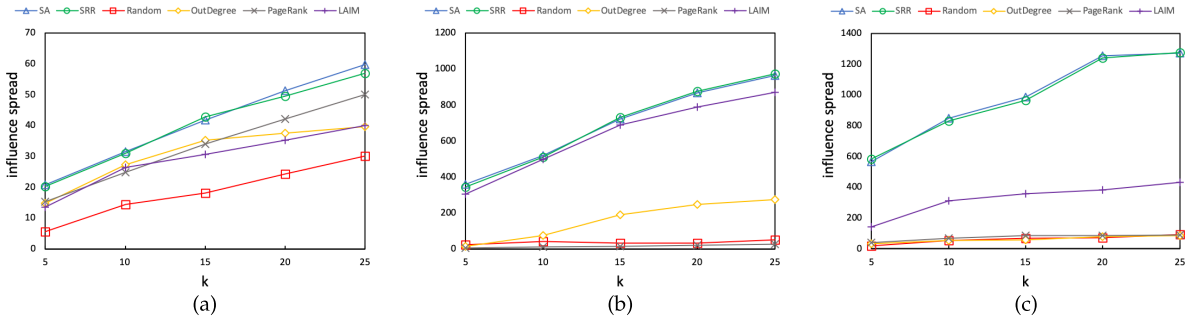

Fig. 3. Influence spread by varying budget k in case-1. (a) Netscience. (b) HepTh. (c) Stanford.

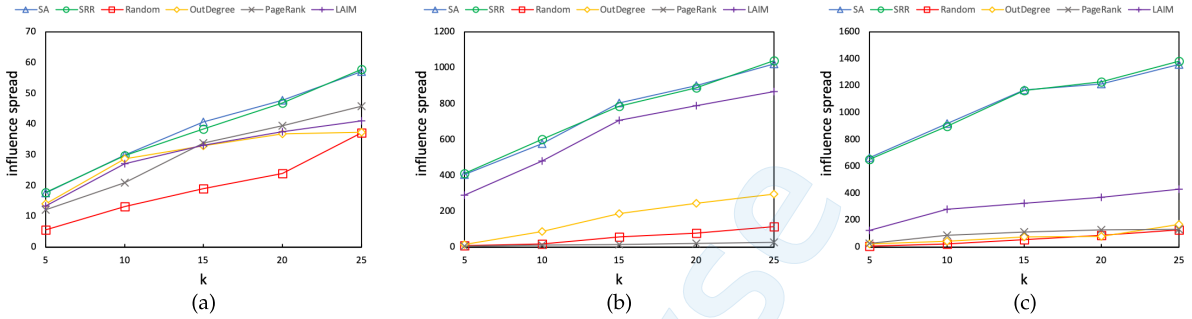

Fig. 4. Influence spread by varying budget k in case-2. (a) Netscience. (b) HepTh. (c) Stanford.

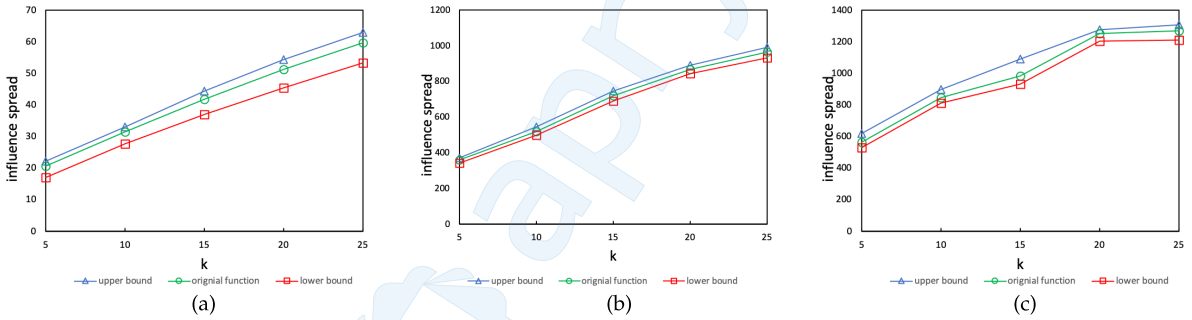

Fig. 5. Influence spread of upper bound, original function, and lower bound in case-1. (a) Netscience. (b) HepTh. (c) Stanford.

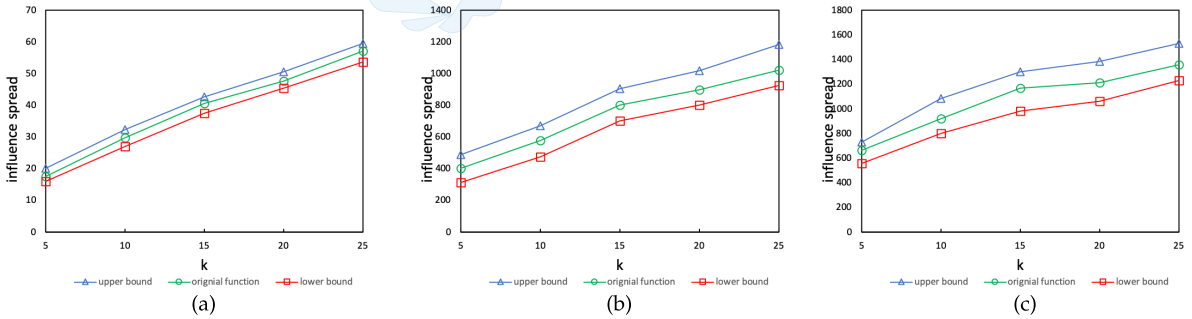

Fig. 6. Influence spread of upper bound, original function, and lower bound in case-2. (a) Netscience. (b) HepTh. (c) Stanford.

matter what the budget k. Take HepTh as an example. For any fixed k, the ratio of the upper bound and original function is about 0.96 in case-1. The ratio of the original function and lower bound is about 0.96 as well. Meanwhile, the ratios are both about 0.85 in case-2 for any fixed k. According to Figs. 5 and 6, we can see that the ratio in case-2 is larger than case-1 in both HepTh and Stanford networks. That is because the 20 influential nodes can influence more nodes than 200 random nodes in HepTh and Stanford networks.

3) Performance With Different Set $\hat{S}_{\mathcal{A}}$: In the following, we focus on the performance when giving different sets $\hat{S}_{\mathcal{A}}$. To reflect the impact of $\hat{S}_{\mathcal{A}}$, we only run algorithms

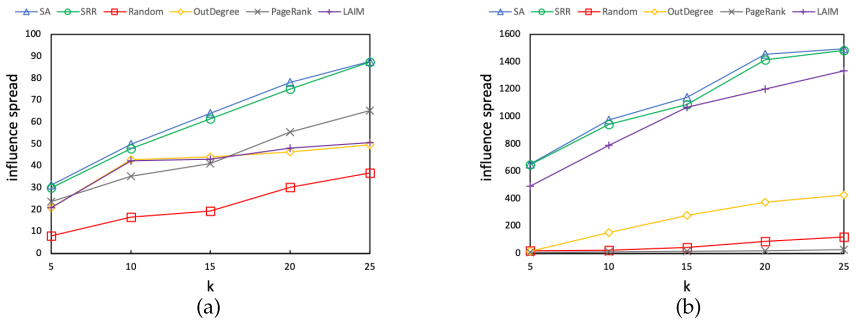

Fig. 7. Influence spread by varying budget k. (a) Netscience. (b) HepTh.

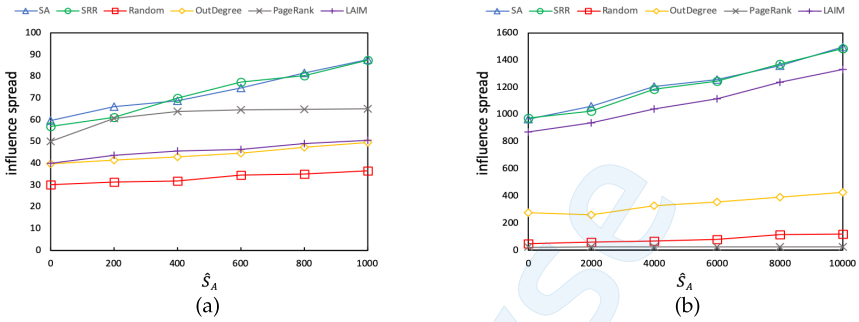

Fig. 8. Influence spread by varying the size of $\hat{S}_{\mathcal{A}}$ when $k = 25$. (a) Netscience. (b) HepTh.

TABLE II
PERFORMANCE WITH DIFFERENT β IN THE NETSCIENCE

Name	$\beta = 2$	$\beta = 3$	$\beta = 4$	$\beta = 5$	$\beta = 6$
k=5	20.75	23.5	25.96	28.23	30.28
k=25	59.74	65.35	72.06	76.83	82.88

TABLE III
RUNNING TIME WHEN $k = 5$ (SECONDS)

Name	case-1		case-2	
	SA	SRR	SA	SRR
Netscience	0.8943	0.9744	0.9463	0.9521
HepTh	43.23	32.03	46.85	38.23
Stanford	517.27	352.91	517.60	346.56

when $S_{\mathcal{A}}$ is in case-1. First, we randomly choose 1000 and 10 000 nodes as $\hat{S}_{\mathcal{A}}$ in the Netscience and HepTh, respectively. As shown in Fig. 7, the influence spread of product $\mathcal{B}$ still increases with the budget k increases. Furthermore, we compare Fig. 7 with Fig. 3. For any fixed k, the results in Fig. 7 are larger than those in Fig. 3. This demonstrates that $\hat{S}_{\mathcal{A}}$ can enforce the final influence spread. Also, we fix $k = 25$ and show the performance with different sizes of $\hat{S}_{\mathcal{A}}$ using Fig. 8. The results returned by the SA and SRR when $k = 25$ are about 50% higher than $k = 5$, respectively.

4) Performance With Different Parameter β: As shown in Table II, we record the influence spread of product $\mathcal{B}$ with different supplementary parameters β in the Netscience. For a fixed k, the influence spread increases with β from 2 to 6. It implies that the supplementary product can increase the influence spread. Moreover, we observe that the increment when $k = 25$ is larger than that when $k = 5$. More specifically, when $k = 5$, the result increases by 10 with β from 2 to 6. And the result increases by 23 when $k = 25$. This is because more seed nodes will influence more nodes.

5) Running Time: We only consider the running time of SA and SRR algorithms since other algorithms are heuristic. Furthermore, we test the running time when $\hat{S}_{\mathcal{A}} = 0$. Table III presents the running time of the SA and SSR algorithms when $k = 5$ on three networks. We observe that the running time increases when the size of the network increases. Meanwhile, the running time of case-1 is close to that case-2. For HepTh and Stanford, the SA algorithm runs longer than the SRR algorithm. This is mainly because the SA algorithm needs to tackle the original function, lower and upper bounds. Finally, the one maximizing the original objective function will be the result. Furthermore, the generation of RR sets is simpler than the generation of SRR sets. Thus, there is not much difference in the running time of the two algorithms in Netscience. Table IV shows the running time of the HepTh by varying the budget k. The running time increases when k increases using both the SA and SRR algorithms.

TABLE IV
RUNNING TIME OF THE HEPTH IN CASE-1 (SECONDS)

Name	$k = 5$	$k = 10$	$k = 15$	$k = 20$	$k = 25$
SA	43.23	60.12	86.38	119.85	182.62
SRR	32.03	42.59	55.10	72.37	105.52

TABLE V
PERFORMANCE WITH $p_{\mathcal{B}}(u,v) \in [0, 0.05]$ IN THE NETSCIENCE

Name	$k = 5$	$k = 10$	$\beta = 15$	$\beta = 20$	$\beta = 25$
SA	10.8	18.75	24.69	33.11	40.21
SRR	8.91	19.24	22.91	31.43	39.49

TABLE VI
PERFORMANCE WITH $p_{\mathcal{B}}(u,v) \in [0, 0.05]$ IN THE NETSCIENCE

Name	$k = 5$	$k = 10$	$\beta = 15$	$\beta = 20$	$\beta = 25$
SA	42.82	69.45	82.17	100.01	116.73
SRR	44.28	70.03	79.24	103.26	114.59

6) Performance With Different Influence Probability: Different from other settings, for each edge (u, v), we sample the influence probability $p_{\mathcal{B}}(u, v)$ from $[0, 0.05]$ and $[0, 0.2]$ in the Netscience, respectively. In the experiments, $p_{\mathcal{A}}(u, v)$ is still sampled from $[0, 0.1]$ and keep constant and $\beta = 2$. As shown in Tables V and VI, experimental results indicate that the result increases when the value range of $p_{\mathcal{B}}(u, v)$ increases. For instance, when $k = 25$, the influence spread is around 40 when $p_{\mathcal{B}}(u, v) \in [0, 0.05]$ and the influence spread is around 115 when $p_{\mathcal{B}}(u, v) \in [0, 0.2]$.

VI. CONCLUSION

This article extends the classic IC to diffuse multiple products and calls this the SIC. Following this model, we study a natural problem, the SIM problem. Given the distribution of the supplementary products, the problem aims at finding k nodes to maximize the influence spread of themselves. We show that the problem is NP-hard, and it is #P-hard to compute the objective function. Fortunately, the objective function is submodular. However, the RIS method cannot apply to estimate the objective function directly due to the impact of the supplementary product. Based on the SA method, we obtain a data-dependent approximate solution. Also, we define the SRR sets and propose an algorithm based on them. At last, to show the effectiveness of our methods, we compare our strategies with some heuristic algorithms on three networks. In the future, we will try to consider the relationships from competition to complementarity based on this model. And we can study the problem in another model.

REFERENCES

[1] P. Domingos and M. Richardson, "Mining the network value of customers," in *Proc. 7th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2001, pp. 57–66.

[2] Y. Zhang, J. Guo, W. Yang, and W. Wu, "Targeted activation probability maximization problem in online social networks," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 1, pp. 294–304, Jan. 2021.

[3] D. Kempe, J. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," in *Proc. 9th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2003, pp. 137–146.

[4] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. Van Briesen, and N. Glance, "Cost-effective outbreak detection in networks," in *Proc. 13th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2007, pp. 420–429.

[5] W. Chen, Y. Wang, and S. Yang, "Efficient influence maximization in social networks," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2009, pp. 199–208.

[6] C. Borgs, M. Brautbar, J. Chayes, and B. Lucier, "Maximizing social influence in nearly optimal time," in *Proc. 25th Annu. ACM-SIAM Symp. Discrete Algorithms*, Jan. 2014, pp. 946–957.

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

[8] H. Wu, J. Shang, S. Zhou, and Y. Feng, "A linear time algorithm for influence maximization in large-scale social networks," in *Neural Information Processing* (Lecture Notes in Computer Science), vol. 10638, D. Liu, S. Xie, Y. Li, D. Zhao, and E. M. El-Alfy, Eds. Guangzhou, China: Springer, 2017, pp. 752–761, doi: 10.1007/978-3-319-70139-4_76.

[9] J. Tang, X. Tang, X. Xiao, and J. Yuan, "Online processing algorithms for influence maximization," in *Proc. Int. Conf. Manage. Data*, May 2018, pp. 991–1005.

[10] J. Guo and W. Wu, "A novel scene of viral marketing for complementary products," *IEEE Trans. Computat. Social Syst.*, vol. 6, no. 4, pp. 797–808, Aug. 2019.

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

[12] H. Zhang, H. Zhang, A. Kuhnle, and M. T. Thai, "Profit maximization for multiple products in online social networks," in *Proc. 35th Annu. IEEE Int. Conf. Comput. Commun. (IEEE INFOCOM)*, Apr. 2016, pp. 1–9.

[13] H. Nguyen and R. Zheng, "On budgeted influence maximization in social networks," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 6, pp. 1084–1094, Jun. 2013.

[14] S. Bharathi, D. Kempe, and M. Salek, *Competitive Influence Maximization in Social Networks*. Berlin, Germany: Springer, 2007.

[15] D. Li, Z.-M. Xu, N. Chakraborty, A. Gupta, K. Sycara, and S. Li, "Polarity related influence maximization in signed social networks," *PLoS ONE*, vol. 9, no. 7, Jul. 2014, Art. no. e102199.

[16] G. Tong, W. Wu, and D.-Z. Du, "Distributed rumor blocking with multiple positive cascades," *IEEE Trans. Computat. Social Syst.*, vol. 5, no. 2, pp. 468–480, Jun. 2018.

[17] Y. Zhang, W. Yang, and D.-Z. Du, "Rumor correction maximization problem in social networks," *Theor. Comput. Sci.*, vol. 861, pp. 102–116, Mar. 2021.

[18] Y. Zhang, X. Yang, S. Gao, and W. Yang, "Budgeted profit maximization under the multiple products independent cascade model," *IEEE Access*, vol. 7, pp. 20040–20049, 2019.

[19] J. Guo, Y. Zhang, and W. Wu, "Overall evaluations on benefits of influence when disturbed by rivals," 2020, *arXiv:2007.01519*.

[20] W. Lu, W. Chen, and L. V. Lakshmanan, "From competition to complementarity: Comparative influence diffusion and maximization," *Proc. VLDB Endowment*, vol. 9, no. 2, pp. 60–71, 2015.

[21] Y. Lin, W. Chen, and J. C. S. Lui, "Boosting information spread: An algorithmic approach," in *Proc. IEEE 33rd Int. Conf. Data Eng. (ICDE)*, Apr. 2017, pp. 883–894.

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

[23] W. Chen, C. Wang, and Y. Wang, "Scalable influence maximization for prevalent viral marketing in large-scale social networks," in *Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2010, pp. 1029–1038.

[24] W. Chen, Y. Yuan, and L. Zhang, "Scalable influence maximization in social networks under the linear threshold model," in *Proc. IEEE Int. Conf. Data Mining*, Dec. 2010, pp. 88–97.

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

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

[27] J. Tang et al., "Efficient approximation algorithms for adaptive seed minimization," in *Proc. Int. Conf. Manage. Data*, Jun. 2019, pp. 1096–1113.

[28] G. Tong, R. Wang, and Z. Dong, "On multi-cascade influence maximization: Model, hardness and algorithmic framework," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 2, pp. 1600–1613, Apr./Jun. 2019.

[29] A. Borodin, Y. Filmus, and J. Oren, "Threshold models for competitive influence in social networks," in *Proc. Int. Conf. Internet Netw. Econ.*, 2010, pp. 539–550.
[30] V. V. Vazirani, *Approximation Algorithms*. Berlin, Germany: Springer-Verlag, 2003.
[31] M. E. J. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 74, no. 3, 2006, Art. no. 036104.
[32] J. Leskovec and A. Krevl. (Jun. 2014). *SNAP Datasets: Stanford Large Network Dataset Collection*. [Online]. Available: http://snap.stanford.edu/data
[33] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank citation ranking: Bringing order to the web," Stanford InfoLab, Stanford, CA, USA, Tech. Rep. 1999-66, Nov. 1999. [Online]. Available: http://ilpubs.stanford.edu:8090/422/

Yapu Zhang received the B.S. degree in mathematics and applied mathematics from Northwest University, Xi'an, Shaanxi, China, in 2016, and the Ph.D. degree in operational research from the University of Chinese Academy of Sciences, Beijing, China, in 2021.

She is currently with the Institute of Operations Research and Information Engineering, Beijing University of Technology, Beijing. Her research interests include social networks, combinatorial optimization, and algorithm design.

Jianxiong Guo (Member, IEEE) received the B.E. degree from the School of Chemistry and Chemical Engineering, South China University of Technology, Guangzhou, China, in 2015, and the Ph.D. degree from the Department of Computer Science, University of Texas at Dallas, Richardson, TX, USA, in 2021.

He is currently an Assistant Professor with the Advanced Institute of Natural Sciences, Beijing Normal University, Beijing, China, and also with the Guangdong Key Laboratory of AI and Multi-Modal Data Processing, Beijing Normal University-Hong Kong Baptist University United International College, Zhuhai, China.

Dr. Guo is a member of Association for Computing Machinery (ACM) and China Computer Federation (CCF).

Wenguo Yang received the M.A. degree in operation research and control theory from Beijing Jiaotong University, Beijing, China, in 2003, and the Ph.D. degree from the University of Chinese Academy of Sciences, Beijing, in 2006.

He is currently a Professor with the School of Mathematics, University of Chinese Academy of Sciences. His research interests include social networks, robust optimization, nonlinear combinatorial optimization, emergency management, and telecommunication network optimization.

Weili Wu (Senior Member, IEEE) received the M.S. and Ph.D. degrees from the Department of Computer Science, University of Minnesota, Minneapolis, MN, USA, in 1998 and 2002, respectively.

She is currently a Full Professor with the Department of Computer Science, University of Texas at Dallas, Richardson, TX, USA. She is involved in the design and analysis of algorithms for optimization problems that occur in wireless networking environments and various database systems. Her current research interests include data communication and data management.