A Novel Scene of Viral Marketing for Complementary Products

Jianxiong Guo and Weili Wu, *Member, IEEE*

***Abstract*— Viral marketing, the method of using a small set of users in social networks to propagate products through cascades, is a well-known and extreme research problem in recent years. Then, influence maximization (IM) is formulated, which aims to select the most influential seeds to maximize the expected total adoption eventually. IM expresses viral marketing perfectly. However, almost all prior work focused on cardinality constraint or considers only simple comparative products model. They neglected that composite complementary products (CCP) are widespread. In other words, when a customer adopts products A and B at the same time, it is possible for him to adopt product C. Therefore, we design a multi-layer network model under independent cascade (IC) model to adapt to multiple complementary products and define the seed selection problem for complementary products model [IM for complementary products (IMCP)] and CCP model (IMCCP) under knapsack constraint. Here, the seed for different products has a different cost. In this paper, we propose two efficient techniques to solve IMCP problem, called Greedy and general-TIM. The Greedy uses simple Greedy Hill-Climbing algorithm under knapsack constraint and obtain $(1/2)(1-(1/e))$-approximation, but the time complexity is hard to accept. The second algorithm, general-TIM, forms a weighted set cover problem by means of randomized sampling (close to Greedy in practice), which reduces the time-consuming significantly. For IMCCP problem, it is difficult to handle because no ready-made algorithms exist to optimize a function that is nonsubmodular and nonsupermodular. Then, we need to get help from sandwich method by finding an upper and lower bound. Finally, our algorithms are evaluated on several real data sets, which prove the correctness of our algorithms.**

***Index Terms*— Composite complementary products (CCP), influence maximization (IM), knapsack, sandwich approximation, social network, viral marketing.**

I. Introduction

THE online social medias, such as Facebook, Twitter, Flickr, Google++, and Linkedln, were booming rapidly in last decades, which provided a platform for communication for many people, and the opportunities were provided by the applications of online social networks for fast information propagation. Viral marketing uses public enthusiasm and interpersonal network to make marketing information spread like a virus. Marketing information is quickly copied to tens of thousands and millions of viewers, penetrate the human brain like a virus, and spread information to more audiences in a short time. Viral marketing was first introduced into public eyes by Domingos and Richardson [1], [2], which resulted in a large number of product adoptions eventually by means of giving free or coupon samples to most influential customers.

Motivated by the notable effect of viral marketing on product adoptions, influence maximization (IM) occurred as a widespread problem about the dissemination of trust, advertisements, or innovations through social graph [3]–[5]. The IM problem was formulated formally by Kempe *et al.* [3] as a discrete optimization problem. Given a directed graph $G = (V, E)$, V is users set and E is the relationship of users, and a positive integer k, IM selects a seed set S^* of k nodes from V to make the expected spread of influence $\sigma(S)$ maximized under a given model m. There were two classical dissemination models [3], accepted by the most researcher, called independent cascade (IC-model) and the linear threshold (LT)-model. The details of these two models will be described in Section III.

$\sigma_m(S)$ can be regarded as the expected number of activated nodes under the model m after the termination of propagation under the seed set S, namely, no new nodes are activated in this step. Under both IC and LT models, the expected spread of influence function $\sigma_m(S)$ is monotone and submodular [3]. A set function $f : 2^V \to R$ is monotone if $f(S) \leq f(T)$ for any $S \subseteq T \subseteq V$. A set function f is submodular if $f(S \cup \{u\}) - f(S) \geq f(T \cup \{u\}) - f(T)$ for any $S \subseteq T \subseteq V$ and $u \notin V \setminus T$. If we can know that a function has the property of monotonicity and submodularity, we can optimize it easily with the help of existing theory. Therefore, it is important for us to find a submodular objective function to solve the problems related to IM.

Even if IM has been studied extremely, a majority of the previous work focused on studying IM problem for single diffusion, which means that there is only one targeted product. However, it is impossible for a company to popularize only one product through social networks. Instead, they always supply their customers with products from different production line to meet the demands of different users. For example, Apple produces both cheaper iPhone 7 and expensive iPhone X; Intel produces ordinary CPU for personal computers and high-speed CPU for servers. Sometimes, people tend to select multiple different products at the same time, typically fast-moving daily necessities like food and clothes. For some products that can be used for a long time, people still like to adopt more because of different function or appearance.

Manuscript received January 31, 2019; revised May 17, 2019; accepted June 27, 2019. Date of publication July 16, 2019; date of current version August 8, 2019. This work was supported by National Science Foundation under Grant 1747818. *(Corresponding author: Jianxiong Guo.)*

The authors are with the Department of Computer Science, Erik Jonsson School of Engineering and Computer Science, University of Texas at Dallas, Richardson, TX 75080 USA (e-mail: jianxiong.guo@utdallas.edu).

Digital Object Identifier 10.1109/TCSS.2019.2926112

Moreover, there is a special kind of product portfolio, called complementary products. Complementary products are those that tend to be purchased together. If a user uses the product A, he/she is very likely to adopt the product B recently, such as iPhone and its accessories, computer hardware and software, and so on. Therefore, from the inspect of companies, viral marketing of complementary products is of great importance. Given a limited budget and a certain number of complementary products with different costs, it is a realistic problem for a company to know how to allocate the budget for the sake of maximizing influence spread.

The comparative IC (Com-IC) model, which combines both competitive and complementary products, was put forward by Lu *et al.* [6]. It told us the diffusion of multiple products in which competitive and complementary relationships exist simultaneously. However, this model is too complex and only suitable for two products to extend to multiple products. In order to design an optimal viral marketing strategy under a budget constraint, we construct a scene of viral marketing for complementary products (VMCP) to simulate that multiple complementary products propagate on social networks. We formulate IM for complementary products (IMCP) problem, which seeks for initial seeds set under limited budget to achieve the goal of maximizing the complementary products influence. The difficulty of solving IMCP is from two aspects. First, the budget needs to be allocated to each product, which is called knapsack constraint, and the existing solution for IM for a single product cannot be applied directly to our case. Second, Monte Carlo (MC) method to estimate the expected influence spread is time-consuming, there are not existing randomized sampling algorithm to solve IM under knapsack constraint. This prompted us to find ways to solve these problems. In addition, considering a more special happening, If a user uses the products A and B, he/she is very likely to adopt the product C recently. We call the tuple (A, B, C) as composite complementary products (CCP), which means that products A and B together can increase the possibility of purchasing product C. For example, if a customer buys a computer and a monitor at the same time, he/she is very likely to buy relative after-sales service. Then, we formulate IMCCP problem, which seek for an initial seeds set under limited budget to achieve the goal of maximiziong the CCP. The difficulty of solving IMCCP is that the objective function is not submodular and supermodular because of existing hyperedge; thus, we need to solve this problem by developing lower bound and upper bound, and then use sandwich approximation framework. Our contributions in this paper are summarized as follows.

1) This is the first attempt to study the viral marketing for multiple complementary products and CCP in social networks. More than one different diffusion propagates upon the networks, limited budget and different costs for different products are considered.
2) We propose a Greedy algorithm for IMCP, which maintains a approximation ratio of $(1/2)(1-(1/e))$ to the optimal solution. Even if the existing technique can be used to improve running time, it is hard to be applied to large-scale networks. We propose the general-TIM algorithm, a weighted set cover problem by means of randomized sampling, which improve its scalability.
3) We solve IMCCP problem, whose objective function is not submodular and supermodular, by sandwich approximation framework.
4) Our algorithms are evaluated on two real-world data sets. The results show both Greedy and general-TIM are better than heuristics, and the influence of Greedy and general-TIM is extremely close. In addition, Sandwich framework is a valid approximation for IMCCP problem.

Organization: Section II discusses related work. Section III describes the VMCP model and defines IMCP problem. Section IV presents the algorithms for IMCP problem. Section V defines IMCCP problem and prove related properties. Section VI describes sandwich approximation framework to solve IMCCP problem. Section VII discusses experiments, and Section VIII is the conclusion.

II. Related Work

Viral marketing was first proposed by Domingos and Richardson [1], [2], and Markov random fields (MRFs) were used to simulate the process of viral marketing. Kempe *et al.* [3] studied IM as a discrete optimization problem. They proved IM to be NP-hard under both IC-model and LT-model and obtained a $(1-(1/e))$-approximation solution by a natural Greedy algorithm [7]. The Greedy algorithm [7] begins with an empty set and selects the node with the largest marginal gain in each step if $|S| \leq k$.

However, it is #P-hard, proved by Chen *et al.* [8], to compute the expected influence $\sigma(S)$ of a seed set S under the IC-model. Thus, a common method to calculate the expected influence $\sigma(S)$ is MC simulation. Then, the approximation ratio for IM drops to $(1-(1/e)-\varepsilon), \varepsilon > 0$. Chen *et al.* [9] analyzed the relation between the number of round r for MC and relative error ε; therefore, we obtain the theoretical basis for the number of r to let ε be small enough. However, most existing methods based on simulation are too slow for large-scale social networks. Some people attempted to reduce the number of MC simulation. Leskovec *et al.* [10] proposed an improved method called CELF, which estimate the upper bounds of influence function because of its submodularity. Most nodes with few influences will not be considered in the later iteration. CELF++, proposed in [11], improve CELF to get better time complexity.

Although there are ways to improve MC, it is difficult to achieve the desired effect. TIM/TIM+ [12] and IM with martingales (IMM) [13] occurred, which makes the IM being scalable under the premise of guaranteeing the approximate ratio. These methods are based on the reverse influence sampling (RIS), proposed by Borgs *et al.* [14], and determine the number of the random reverse reachable set (RR-set) needed to ensure approximation ratio. It required OPT, the optimal expected influence of valid seed set, to estimate the number of RR-set. However, OPT is difficult to determine, Tang *et al.* [12] proposed a bunch of parameter estimation technique to estimate OPT. Then, IMM appeared, which uses a martingale analysis to estimate OPT more efficiently. This better parameter estimation improves TIM/TIM+.

Later, Lu *et al.* [6] proposed the comparative IC (Com-IC), which combines both competitive and complementary products. There are two problems of how to maximize the own and incremental influence to others defined in [6]. The problem mentioned earlier is NP-hard, and they used a method similar to TIM to solve the two problems. Interactive LT (ILT) model was proposed by Ou *et al.* [15], who adapted LT-model to the scene that multiple diffusions exist.

Although there is a large number of existing work for IM problem, nearly all of them considered the optimization problem with submodularity. A few methods can be used directly for nonsubmodular optimization problem. For monotone nonsubmodular maximization, there are several choices of methdology, such as supermodular degree [16]–[18], sandwich approximation framework [6], [19], [20], and algorithms based on difference of submodular function decomposition: submodular–supermodular algorithm [21], modular–modular algorithm [22], and iterated sandwich method [23].

III. Problem Definition

In this section, we give the preliminaries, including influence model and notation, to this paper, then the problem is formulated.

A. Influence Model

A social network can be expressed as a directed graph $G' = (V', E')$, generally, the users are denoted as V' and edge $e = (u, v) \in E'$ denotes the relationship between user u and user v. The number of nodes and edges in graph G' is n and m, respectively. The set of incoming neighbors and outgoing neighbors of node v is denoted as $N^-(v)$ and $N^+(v)$, respectively. Let node set and edge set in the directed graph G' be denoted as $V(G')$ and $E(G')$, respectively. In order to represent the spreading of new information or technology, Kempe *et al.* [3] proposed two classical diffusion model, IC-model and LT-model.

1) Independent Cascade Model: Each node v is attempted to be activated independently by its incoming neighbors $N^-(v)$, and the activation probability is $p_{uv}, u \in N^-(v)$. Given an activation probability p_{uv} for each pair of edges (u, v), the propagation process can be described in discrete rounds. In round t, each node u activated in round $t-1$ will has one chance to activate the nodes in its outgoing neighbors $N^+(u)$, which is inactive in round t, with activation probability p_{uv}. Then, the propagation process terminates when there are no nodes become active in this round.

2) Linear Threshold Model: For each edge $e = (u, v) \in E(G)$, a weight b_{uv} is correlated with it. Each node $v \in V(G)$ satisfies that $\Sigma_{u \in N^-(v)} b_{uv} \leq 1$. In addition, each node $v \in V(G)$ is correlated with a threshold θ_v, which is uniformly distributed in interval [0, 1]. Given that, the propagation process can be described in discrete rounds. In round t, the nodes that have been activated in round $t-1$ are still active. Any inactive node v will become active if the total weight associated with active nodes in its incoming neighbors $N^-(v)$ is greater than θ_v. Then, the propagation process terminates when there are no nodes become active in this round.

Although both IC-model and LT-model can be applied to our problem, LT-model is hard to construct multi-layer network later, because we need to add edges among different layers, it is difficult to define weight corresponding to these edges, which will be clear after defining our problem formally. Thus, our diffusion model is based on IC-model.

B. Problem Definition

Assuming that there are t products, we are interested in a problem with viral marketing for these t products. Considering t products $h = \{h_1, h_2, \ldots, h_t\}$ with costs $c = \{c_1, c_2, \ldots, c_t\}$, respectively, there are some complementary products in h, which means that influence propagation for any two complementary products happen dependent of each other, in other words, a customer u who adopts product h_i, may adopt product h_j, $h_j \in h \setminus h_i$ as well, and that a customer can get influenced by any number of products. Here, we suppose that a node u who adopts product h_i would also adopt product h_j with probability p_{ij}, where $i \in \{1, 2, \ldots, t\}$ and $j \in \{1, 2, \ldots, t\}$. We define the VMCP as follows.

Definition 1 VMCP: Given a budget B, find a set of customers for giving free samples within the budget B to maximize the expected total sales of the product. For this problem, there are two types of influences to each node v for each product h_j at each step of the information diffusion process as follows.

1) The influence from a node u who adopts product h_j and has the probability p_{uv} for the success of the influence to node v.
2) The influence from a product h_i because node v adopts h_i at the previous step and has the probability p_{ij} for the success of the influence to product h_j.

Theorem 2: The problem of VMCP in $G' = (V', E')$ is equivalent to weighted IM problem in $G = (V, E)$. Here, for each node u, assigning a cost $c(u)$ to choose u as a seed for influence propagation. Given budget B, find a set of nodes as seeds within the budget B to maximize the expected number of active nodes.

Proof: Let $G' = (V', E')$ be the original social network in the problem of VMCP. For each product h_i, make a copy G^i of G'. Here, we define u^i in G^i is the copy of corresponding node u in G'. Considering the graph $G^1 \cup G^2 \cup \cdots \cup G^t$, for each corresponding nodes u^i and u^j, which come from different layers G^i and G^j, adding an edge (u^i, u^j) associated with a probability p_{ij} if they are complementary. Here, the edge (u^i, u^j) means that customer u is likely to adopt product j with probability p_{ij} after adopting product i. The new adding edge set, we call it as complementary edge set, denoted by CES. Let $G = G^1 \cup G^2 \cup \cdots \cup G^t \cup \text{CES}$, the problem of VMCP in $G' = (V', E')$ would be equivalent to the IM on constructed network $G = (V, E)$ with cost function $c(u^i) = c_i$, $u^i \in V(G^i)$. We take Fig. 1 as an example to show how to construct the new multi-layer network $G = (V, E)$. Fig. 1 shows a realization g of G, which is a subgragh of G where $V(\text{g}) = V(G)$ and $E(\text{g}) \subseteq E(G)$. For each edge $e = (u, v) \in E(G)$, it appears in realization g with probability p_{uv}. There edges appear in realization g are

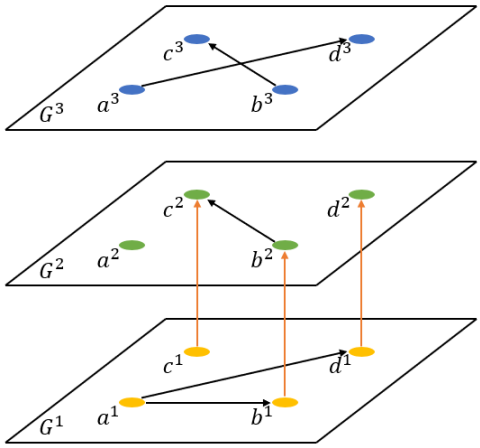

Fig. 1. Form of expression to constructed graph, where each layer represents one product and the nodes in the same column can be connected by complementary probabilities.

referred as to "live" edges, whose propagation probability is equal to 1. The propagation in a realizaiton g is a deterministic process. Considering there are three products $h = \{h_1, h_2, h_3\}$, where h_1 and h_2 are complementary products, we need to add CES from G^1 to G^2, shown as orange arrow in Fig. 1. We select customer a for product h_1 (node a^1) as seed, then the node set being activated is $\{a^1, b^1, d^1, b^2, c^2, d^2\}$. From here, we can see that there are some customers adopting product h_2 because of complementary property, even if we do not select any seeds for product h_2. ■

On constructed graph $G = (V, E)$, the objective is to select the seed set $S = S_1 \cup S_2 \cup \cdots \cup S_t$, where $S_i \subseteq V(G^i) \subseteq V(G)$ is the seed set for product h_i, subject to the budget constraint $\Sigma_{i=1}^{t} c_i |S_i| \leq B$. In addition, $I(S_i)$ is considered as the node set being activated (including the seed S_i) by selecting the seed set S_i. Because the diffusion process is dependent to each different product, the total eventually influenced nodes $|I(S)|$ is not equal to the sum of each separate influenced node $|I(S_i)|$, namely, $|I(S)| \neq \Sigma_{i=1}^{t} |I(S_i)|$, because of $I(S_i) \cap I(S_j) \neq \emptyset$. Therefore, we define the total eventually influenced nodes as $I(S) = \cup_{i=1}^{t} I(S_i)$ and define $\sigma(S)$ as its expected value. We aim to select the best S to maximize this expected influence $\sigma(S)$. We are now ready to define the IMCP on constructed graph G as follows.

Definition 3 IMCP: Given the constructed graph $G = (V, E)$, costs $c = \{c_1, c_2, \ldots, c_t\}$ and budget B, find the seed set $S = S_1 \cup S_2 \cup \cdots \cup S_t$, where $S_i \subseteq V(G^i) \subseteq V(G)$, which satisfies budget constraint, $\Sigma_{i=1}^{t} c_i |S_i| \leq B$, and maximize $\sigma(S)$. Here, we define total cost as $c(S) = \Sigma_{i=1}^{t} c_i |S_i|$, so budget constraint is $c(S) \leq B$.

It is obvious to see that the IMCP is a generalization of standard IM problem in [3], when setting $c_1 = c_2 = \cdots = c_t = 1$. Hence, this problem is NP-hard, monotone nondecreasing, and submodular, which can be classified as a monotone submodular maximization problem with knapsack constraint. Nemhauser *et al.* [7] proved that the simple Greedy algorithm has a performance guarantee $(1 - (1/e))$ with cardinality constraint.

IV. Solution for IMCP

From above, we have known that IMCP is a nondecreasing submodualr maximization problem under the knapsack constraint. In this section, we want to know how to solve it efficiently.

A. Simple Greedy Algorithm

We now introduce our first technique, call simple Greedy, to solve the IMCP problem. The main idea is to use simple Hill-Climbing algorithm to the complementary products, subject to the knapsack constraint. At each step, it selects a node u^i from $V(G^i)$ such that adding u^i to S_i give the maximum increase $(\sigma(S)/c_i)$ to overall influence. We repeat this until it violates the knapsack constraint. The pseudocode of simple Greedy algorithm is shown in Algorithm 1.

Algorithm 1 Simple Greedy (G, c, B)

Input: Graph $G = (V, E)$, costs c and budget B
Output: Seed set S

```
Initialize S ← ∅
while V ≠ ∅ do
    select u^i from V(G^i) that maximize (σ(S ∪ {u^i}) − σ(S))/c_i)
    if c(S) + c_i ≤ B then
        S_i ← S_i ∪ {u^i}
    end if
    V ← V \ {u^i}
end while
return S
```

It is obvious that the simple Greedy that selects at each step, a node u^i that maximize $(\sigma(S \cup \{u^i\}) - \sigma(S)/c_i)$ has an unbounded approximation factor. For example, there are two products together, $c_1 = 1$ and $c_2 = p + 1$, and $p_{12} = 0$. Considering an realization g of the constructed graph G, we assume that there are no edges in the subgraph G^1, namely, $E(G^1) = \emptyset$, and exists a node u^2 in subgraph G^2 such that $\sigma(u^2) = p$ and other nodes $v^2 \in V(G^2) \setminus u^2$, $\sigma(v^2) < p$. The budget $B = p + 1$. The optimal solution should be u^2 and has influence p, while the solution selected by simple Greedy algorithm should be a node u^1 from $V(G^1)$ and has influence 1, because of $(\sigma(u^1)/c_1) > (\sigma(u^2)/c_2)$. The approximation ratio for this example $1/p$. It is extremely bad when $p \to \infty$. Therefore, the simple Greedy algorithm is not suitable for us to solve IMCP problem, however, we are able to revise it slightly to get avoid extreme bad happening and get a valid approximation factor.

B. Greedy Algorithm

Although extremely bad approximation ratio may be occurs in simple Greedy algorithm, a small modification can achieve a constant approximation ratio for a nondecreasing submodular function under the knapsack constraint. Khuller *et al.* [24] proposed a Greedy algorithm with $(1/2)(1 - (1/e))$-approximation, whose first part is the same as

simple Greedy algorithm, but they select a node u that maximize $(\sigma(S \cup \{u^i\}) - \sigma(S)/c_i)$ later, compare with the result of simple Greedy algorithm and then select the larger one. Leskovec *et al.* [10] proposed another effective methods, called cost-effective forward (CEF) selection to get the same approximation. They compute the solution A_{GBC} using simple Greedy algorithm and also compute A_{GUC} using standard Greedy algorithm within cardinality constraint (setting $c_i = 1$). CEF returns the better solution and state that $\max\{\sigma(A_{GCB}), \sigma(A_{GUC})\} \geq (1/2)(1-(1/e))\cdot\text{OPT}$. However, our Greedy algorithm is slightly different with above, which learn from Algorithm 1-B in [25]. From [25], knapsack problem can be formulated as a $0-1$ integer programming problem: given n items with volume s_i and value c_i, we aim to maximize $\sum_{i=1}^{n} c_i x_i$ subject to $\sum_{i=1}^{n} s_i x_i \leq S$ and $x_i \in \{0, 1\}$, where $1 \leq i \leq n$. Greedy algorithm can be described as:

1) sorting all items in the nonincreasing order of c_i/s_i, we can assume that $c_1/s_1 \geq c_2/s_2 \geq \cdots \geq c_n/s_n$ holds;
2) if $\sum_{i=1}^{n} s_i \leq S$ then output $c_G \leftarrow \sum_{i=1}^{n} c_i$, else $k \leftarrow \max\{j | \sum_{i=1}^{j} s_i \leq S \leq \sum_{i=1}^{j+1} s_i\}$. Eventually, output $c_G \leftarrow \max\{c_{k+1}, \sum_{i=1}^{k} c_i\}$.

Du *et al.* [25] have proven that c_G is a two-approximation solution to knapsack problem. It is easy to extend this idea to solve nondecreasing submodualar function under the knapsack constraint. The pseudocode of the Greedy algorithm is shown in Algorithm 2.

Algorithm 2 Greedy (G, c, B)

Input: Graph $G = (V, E)$, costs c and budget B
Output: Seed set S

```
Initialize S ← ∅
Initialize last_item ← ∅
while c(S) ≤ B do
    select u^i from V(G^i) that maximize (σ(S ∪ {u^i}) − σ(S))/c_i)
    S_i ← S_i ∪ {u^i}
    V ← V \ {u^i}
    last_item ← {u^i}
end while
result ← S \ last_item
return arg max{σ(result), σ(last_item)}
```

Theorem 4: Greedy algorithm has an approximation of $(1/2)(1-(1/e))$ for nondecreasing submodular maximization under the knapsack constraint.

Proof: Let $\{v_1, v_2, \ldots, v_{k+1}\}$ be generated by Algorithm 2 and denote $A_i = \{v_1, v_2, \ldots, v_i\}$. Then, $v_{i+1} = \arg\max_{v \in \{V \setminus A_i\}} \Delta_v f(A_i)/c(v)$ and $c(A_k) < B \leq c(A_{k+1})$. Here, we define $\Delta_v f(A_i) = f(A_i \cup \{v\}) - f(A_i)$. Given set S, $T = \{t_1, t_2, \ldots, t_n\}$, and $T_i = \{t_1, t_2, \ldots, t_i\}$, it is easy to know that

$$f(S \cup T) = f(S) + \sum_{i=1}^{n} \Delta_{t_i} f(S \cup T_{i-1}). \tag{1}$$

Therefore, supposing optimal $A^* = \{u_1, u_2, \ldots, u_h\}$, we have $f(A^*) \leq f(A_i \cup A^*) = f(A_i) + \Delta_{u_1} f(A_i) + \Delta_{u_2} f(A_i \cup \{u_1\}) + \cdots + \Delta_{u_h} f(A_i \cup \{u_1, \ldots, u_{h-1}\}) \leq f(A_i) + \Delta_{u_1}(A_i) + \Delta_{u_2} f(A_i) + \cdots + \Delta_{u_h} f(A_i) \leq f(A_i) + (c(u_1)/c(v_{i+1}))\Delta_{v_{i+1}} f(A_i) + (c(u_2)/c(v_{i+1}))\Delta_{v_{i+1}} f(A_i) + \cdots + (c(u_h)/c(v_{i+1}))\Delta_{v_{i+1}} f(A_i) = f(A_i) + (c(A^*)/c(v_{i+1}))(f(A_{i+1}) - f(A_i))$.

Then, we denoting $\alpha_i = f(A^*) - f(A_i)$, and from above, we know that $\alpha_i \leq (c(A^*)/c(v_{i+1}))(\alpha_i - \alpha_{i+1})$. We have

$$\alpha_{i+1} \leq \left(1 - \frac{c(v_{i+1})}{c(A^*)}\right)\alpha_i \leq e^{-c(v_{i+1})/B}\alpha_i \tag{2}$$

since $1 + x \leq e^x$ for any real number x and $c(A^*) \leq B$. Thus, we have $\alpha_{k+1} \leq e^{-c(v_{k+1})/B}\alpha_k \leq e^{-c(v_{k+1}+v_k)/B}\alpha_{k-1} \leq e^{-c(A_{k+1})/B}\alpha_0 \leq e^{-1} f(A^*)$ recursively. Now, we know that $f(A^*) - f(A_{k+1}) \leq e^{-1} f(A^*)$ and $f(A_{k+1}) \geq (1 - e^{-1}) \cdot f(A^*)$. Since $f(A_k) + f(\{v_{k+1}\}) \geq f(A_{k+1}) + f(\emptyset) \geq f(A_{k+1})$ by submodular property, we have $\max\{f(A_k), f(\{v_{k+1}\})\} \geq (1 - e^{-1})/2 \cdot f(A^*)$. ■

Even though there are algorithm with better approximation factor existing, they are at the expense of consuming more time. For example, Algorithm 2 by Khuller *et al.* [24] proved that it is a $(1 - 1/e)$ approximation algorithm to solve maximum coverage problem, which use the enumeration technique. Let d be some fixed integer number, we consider all the subsets of V with cardinality d under the constraint of budget B, then continue to compute each of these subsets by Greedy method. In reality, the extremely bad case hardly appears unless the difference of cost for different products is very large. Thus, the Greedy algorithm is enough to get a satisfactory solution in most case.

C. General-TIM Algorithm

In Section III, we have shown that the objective function σ of IMCP is nondecreasing and submodular under the knapsack constraint, thus the Greedy algorithm can get a $(1/2)(1 - (1/e))$-approximation solution. However, the computational cost of Greedy algorithm is too high because the objective function $\sigma(S)$ is #P-hard to compute given seed set S, and the usual method to compute is MC simulation. The time complexity of the Greedy algorithm is $O((\max_{c_i \in c} \lfloor (B/c_i) \rfloor)mnr)$, where r is simulation times using a MC method. This drives us to design a more desirable approach, which can obtain a similar solution set in a timely manner.

For IM, Tang *et al.* [12] proposed the two-phase IM (TIM) algorithm that produces a $(1 - 1/e - \varepsilon)$-approximation with at least $(1 - n^{-\ell})$ probability in $O((k + \ell)(m + n)\log n \cdot \varepsilon^{-2})$. It is based on a technique called RIS. First, we need to introduce two important concepts, RR-set and random RR-set, proposed by Brogs *et al.* [14]. Given a realization g of G and a node v in g, the RR-set is a set in which all the nodes in g can reach v. The random RR-set is an RR-set generated on a realization g sampled from the distribution of realization, and a node which is selected from $V(G)$ randomly. In TIM algorithm, we need to obtain a certain number of random RR-set. Then, in the problem of IM, for a given node, if it appears more times in these random RR-sets, this

Algorithm 3 General-TIM (G, c, B)

Input: Graph $G = (V, E)$, costs c and budget B
Output: Seed set S

```
Initialize a set R = ∅
Generate θ random RR-set R
Initialize S ← ∅
Initialize last_item ← ∅
while c(S) ≤ B do
    Select u^i from V(G^i) that maximize (1/c_i)·(the number of RR-sets in R covered by u_i)
    S_i ← S_i ∪ {u^i}
    V ← V \ {u^i}
    R ← R \ {RR-sets covered by u_i}
    last_item ← {u^i}
end while
result ← S \ last_item
return arg max{σ(result), σ(last_item)}
```

node has a larger influence than others with high probability. We can expand this technique to solve our IMCP problem under knapsack constraint, the pseudocode of general-TIM is shown in Algorithm 3. Let $F_{\mathcal{R}}(S)$ be the fraction of RR-sets in $\mathcal{R}$ covered by S, we have [12]

$$\mathbb{E}[nt \cdot F_{\mathcal{R}}(S)] = \sigma(S) \tag{3}$$

where t is the number of different products. Now, IMCP probem can be transformed into the budgeted maximum coverage problem and general-TIM produces a $(1/2)(1-(1/e))$-approximation to the solution that maximizes the coverage of $\mathcal{R}$ under the cost constraint, because $F_{\mathcal{R}}(S)$ is nondecreasing and submodular with respect to S.

Next, we need to discuss how to estimate θ so that it is large enough to ensure that our solution is accurate with high probabiltiy. We can consider that the maximum number of selected nodes k from the nodes of constructed graph $V(G)$ under limited budget B is $\max_{c_i \in c}\lfloor (B/c_i)\rfloor$, in other words, we select the nodes with minimum cost as seeds, then we can get the most number of seeds. Thus, we have the following.

Lemma 5: Define $k = \max_{c_i \in c}\lfloor (B/c_i)\rfloor$ and suppose $\theta \geq \lambda/\text{OPT}$, where

$$\lambda = \frac{(8+2\varepsilon)(nt)\cdot(\ell\log(nt) + \log\binom{nt}{k} + \log 2)}{\varepsilon^2} \tag{4}$$

and t is the number or different products. Then, for any seed set S under the limited budget B, $|nt \cdot F_R(S) - \sigma(S)| \leq (\varepsilon/2)\cdot \text{OPT}$ holds with at least $1-(nt)^{-\ell}/\binom{nt}{k}$ probability.

Proof: The proof is very similar to [12, Lemma 3] by means of Chernoff bound, but $k = \max_{c_i \in c}\lfloor (B/c_i)\rfloor$ and nt instead of n. The reason for the correctness of the total number of nodes in G is nt. IMCP problem is equivalent to IM problem, which has been proven in Theorem 3.1. Here, we need to explain why $\binom{nt}{k}$ happens. Because it is NP-hard to find the number of possible seed sets under the knapsack constraint, we attempt to find an upper bound for the number of valid seed sets. Considering constructed network G, there are nt nodes, and under budget constraint, the number of nodes in a valid seed set cannot be larger than k. Thus, the number of valid seed sets is at most $\binom{nt}{k}$, which is an upper bound of the number of valid seed sets. ■

Theorem 6: Given θ that satisfies $\theta \geq \lambda/\text{OPT}$, Algorithm 3 returns a $((1/2)(1-(1/e))-\varepsilon)$-approximation solution for IMCP problem with at least $1-(nt)^{-\ell}$ probability.

Proof: Let S_B be the seed set returned by Algorithm 4, and S'_B be the node set that maximizes $F_R(\cdot)$ under limited budget. As $F_R(S_B)$ is a $(1/2)(1-(1/e))$-approximation solution for the budgeted maximum converage problem, we have $F_R(S_B) \geq (1/2)(1-(1/e))\cdot F_R(S'_B)$. Let S''_B is the optimal solution for the IMCP problem on G, hence $\sigma(S''_B) = \text{OPT}$. We have $F_R(S'_B) \geq F_R(S''_B)$, which leads to $F_R(S_B) \geq (1/2)(1-(1/e))\cdot F_R(S''_B)$.

Assume that θ that satisfies $\theta \geq \lambda/\text{OPT}$. From Lemma 5, $|nt \cdot F_R(S) - \sigma(S)| \leq (\varepsilon/2)\cdot \text{OPT}$ holds with at least $1-(nt)^{-\ell}/\binom{nt}{k}$ probability for any valid seed sets under limited budget B. We can assume that the number of valid seed sets under knapsack constraint as β. By the union bound, $|nt \cdot F_R(S) - \sigma(S)| \leq (\varepsilon/2)\cdot \text{OPT}$ should be held for all seed sets under knapsack constraint with at least $1-\beta(nt)^{-\ell}/\binom{nt}{k}$. Because the number of nodes in a valid seed set is at most k, we can know that $\beta \leq \binom{nt}{k}$, so $1-\beta(nt)^{-\ell}/\binom{nt}{k} \geq 1-(nt)^{-\ell}$. In that case, we have

$$\begin{aligned}
\sigma(S_B) &> nt \cdot F_R(S_B) - \frac{\varepsilon}{2}\cdot \text{OPT} \\
&\geq \frac{1}{2}\left(1-\frac{1}{e}\right)\cdot nt \cdot F_R\left(S'_B\right) - \frac{\varepsilon}{2}\cdot \text{OPT} \\
&\geq \frac{1}{2}\left(1-\frac{1}{e}\right)\cdot nt \cdot F_R\left(S''_B\right) - \frac{\varepsilon}{2}\cdot \text{OPT} \\
&= \left(\frac{1}{2}\left(1-\frac{1}{e}\right) - \left(\frac{3e-1}{4e}\right)\varepsilon\right)\cdot \text{OPT} \\
&> \left(\frac{1}{2}\left(1-\frac{1}{e}\right) - \varepsilon\right)\cdot \text{OPT}.
\end{aligned}$$

From above, the theorem is proven, we obtain a theoretical bound for this problem. ■

Even if we get a theoretical bound, we need to estimate the value of OPT. There are some existing methods to estimate the value of OPT, whose ideas are to get a lower bound of OPT. For example, Tang *et al.* [12] stated that KPT $\leq$ OPT, so we can use KPT instead of OPT. Suppose that we select k-size node set according to a probability distribution based on its in-degree for each node, KPT is the mean of expected spread for this node set. In addition, IMM [13] used martingale analysis and a more efficient estimation method on OPT. These methods can be extended to our problem easily. Because it is not the focus of this paper, we will not expand in depth here. We can determine the value of θ by the use of experimental simulation later.

V. Composite Complementary Product

We have introduced the CCP in Section I. If a node adopts the influence of A and B, it has a higher probability to adopt C. We refer to this triad (A, B, C) as a CCP. How can we deal with it in our constructed graph $G = (V, E)$ above if there exists CCP? We formulate our problem as follows.

Given the constructed graph $G = (V, E)$ above, we can obtain hypergraph $\bar{G} = (V, \bar{E})$ by adding some hyperedge into G. The influence from the product h_i and h_j because customer v adopts both h_i and h_j at the previous step and has the probability p_{ij-k} for the success of the influence to adopt product h_k. In constructed graph, for each corresponding nodes u^i, u^j and u^k, which come from different layers G^i, G^j and G^k, adding an hyperedge $(\{u^i, u^j\}, u^k)$ associated with a probability p_{ij-k}. We can obtain the new constructed hypergragh $\bar{G} = (V, \bar{E})$. For a hyperedge $e = (H_e, v)$, the head set is H_e and the tail node is v. $p_{H_e,v}$ is the probability that H_e influences v when all the nodes in head set are active. The propagation process is the same as IC-model. We are now ready to define the IMCCP on new constructed graph $\bar{G}$ as follows.

Definition 7 (IMCCP): Given the constructed hypergraph $\bar{G} = (V, \bar{E})$, costs $c = \{c_1, c_2, \ldots, c_t\}$ and budget B, find the seed set S that satisfies budget constraint, $c(S) \leq B$, and maximize the expected spread $\sigma(S)$.

It is obvious to see that the IMCCP is a generalization of standard IM problem in [3], when setting $c_1 = c_2 = \cdots = c_t = 1$ and no CCP (hyperedge) exists. Hence, this problem is NP-hard, but it is not submodular or supermodular as follows.

Theorem 8: $\sigma(S)$ is not submodular function in hypergraph $\bar{G} = (V, \bar{E})$ under IC-model.

Proof: We prove by a counterexample. Constructed hypergraph $\bar{G} = (V, \bar{E})$ has $V = \{u^1, u^2, u^3\}$ in different layers of $\bar{G}$, $\bar{E} = (\{u^1, u^2\}, u^3)$ and $p_{12-3} = 1$. Let $A = \emptyset$ and $B = \{u^1\}$, we have $\sigma(A) = 0$ and $\sigma(B) = 1$. Putting u^2 into A and B, we have $\sigma(A \cup \{u^2\}) = 1$ and $\sigma(B \cup \{u^2\}) = 3$. Then, $\sigma(A \cup \{u^2\}) - \sigma(B \cup \{u^2\}) < \sigma(A) - \sigma(B)$ when $A \subseteq B$. Thus, $\sigma(S)$ is not a submodular function. ■

Theorem 9: $\sigma(S)$ is not supermodular function in hypergraph $\bar{G} = (V, \bar{E})$ under IC-model.

Proof: We prove by a counterexample. Constructed hypergraph $\bar{G} = (V, \bar{E})$ has $V = u^1, v^1$ in the same layers of $\bar{G}$, $\bar{E} = (u^1, v^1)$ and $p_1 = 1$. Let $A = \emptyset$ and $B = \{u^1\}$, we have $\sigma(A) = 0$ and $\sigma(B) = 2$. Putting v^1 into A and B, we have $\sigma(A \cup \{v^1\}) = 1$ and $\sigma(B \cup \{v^1\}) = 2$. Then, $\sigma(A \cup \{v^1\}) - \sigma(B \cup \{v^1\}) > \sigma(A) - \sigma(B)$ when $A \subseteq B$. Thus, $\sigma(S)$ is not a supermodular function. ■

From above, we prove IMCCP is not submodular or supermoduler. For a nonsubmodular function, there are no general methods to get a solution with a certain approximation ratio. Lu *et al.* [6] provide us with a sandwich approximation solution, which needs us to find the lower bound and upper bound of objective function.

A. Upper Bound

Upper bound can be obtained easily by replacing hyperedge influence with separate node-to-node influence that has the same activation probability with hyperedge influence, which amplifies the activation probability [26]. For constructed hypergraph $\bar{G} = (V, \bar{E})$, consider upper bound constructed graph $\bar{G}_v = (V, \bar{E}_v)$, $\bar{E}_v$ is directed edge set, as follows. Considering node u and node v in $V(\bar{G})$, a directed edge (u, v) with probability p_e is generated if $u \in H_e$ and hyperedge $e = (H_e, v)$ exists. Suppose there are n head set of $(H_1, v), (H_2, v), \ldots, (H_n, v)$ in $\bar{G}$ that contain node u, the new influence probability $p_{uv} = 1 - \prod_{i=1}^{n}(1 - p_{H_i,v})$ [26]. This method is equivalent to connect each node in H_e to v with probability p_e, and then combine multiple edge set M between two nodes to one edge with probability $p = 1 - \prod_{m \in M}(1 - p_m)$. However, it is not the best thought to obtain the upper bound because the activation probability to v will become $1 - (1 - p_e)^{|H_e|}$ when all the node in H_e is activated, which is extremely bad when $|H_e|$ is very large. If the nodes in H_e cannot be activated together eventually, it will be worse than above that all the nodes in H_e are activated.

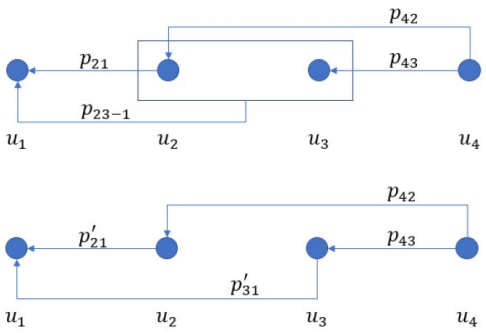

Fig. 2. Example of converting hyperedge to common edges to get upper bound.

From above, we can know that it is not a good idea to connect all nodes in H_e to v with probability p_e. Therefore, we need to minimize the new probability from each node in H_e to v. Instead, we replace each node in H_e to v with a smaller probability $p_{u \in H_e,v} = 1 - (1 - p_e)^{(1/|H_e|)}$, then combine multiple edge set M between two nodes to one edge with probability $p = 1 - \prod_{m \in M}(1 - p_m)$. Then, we can think about the problem like this, each node $u \in H_e$ activates v independently, and the eventual activation probability of v is p_e when all the nodes in H_e have been activated. A clear example is shown in Fig. 2. First, we decompose hyperedge $(\{u_2, u_3\}, u_1)$ with p_{23-1} to directed edge (u_2, u_1) and (u_3, u_1) with $\bar{p}_{21}$ and p'_{31}. Here, $\bar{p}_{21} = p'_{31} = 1 - (1 - p_{23-1})^{1/2}$. Then, merging the multiple edge (u_2, u_1), we have $p'_{21} = 1 - (1 - p_{21})(1 - \bar{p}_{21}) = 1 - (1 - p_{21})((1 - p_{23-1})^{1/2})$. Then, the upper bound problem for IMCCP is formulated.

Definition 10 (Upper Bound Problem): Given the graph $\bar{G}_v = (V, \bar{E}_v)$, $c = \{c_1, c_2, \ldots, c_t\}$ and budget B, find the seed set S that satisfies budget constraint, $c(S) \leq B$, and maximize the expected spread $\nu(S)$.

Here, we transform hypergraph $\bar{G} = (V, \bar{E})$ to directed graph $\bar{G}_v = (V, \bar{E}_v)$, which is equivalent to the constructed graph in IMCP problem; thus, it is nondecreasing and submodular. We can use Algorithm 2 and Algorithm 3 to solve it and get a $(1/2)(1 - (1/e))$ approximation ratio.

Theorem 11: Given $\bar{G}_v = (V, \bar{E}_v)$, $\nu(S)$ is an upper bound of $\sigma(S)$ with respect to seed set S.

Proof: $\nu(S) \geq \sigma(S)$ for $S \subseteq V$ if $\nu(S)$ is an upper bound of $\sigma(S)$. We define that S_t and S_t^v are the activated node in $\bar{G}$ and $\bar{G}_v$ at the time step t. At any time step, S_t should be the subset of S_t^v.

When $t = 0$, we need to prove $S_0 = S_0^v = S = \emptyset \Rightarrow S_1 \subseteq S_1^v$. For each node u which is not activated, we define activation probability $p(u)$ in $\bar{G}$ and $p_v(u)$ in $\bar{G}_v$. Then, the probability that u can be activated in $\bar{G}$ is $p(u) = 1 - \prod_{H_i \subseteq S}(1 - p_{H_i,u})$, where there exists n hyperedges $\{(H_1, u), (H_2, u), \ldots, (H_n, u)\}$ in $\bar{G}$. The probability that u can be activated in $\bar{G}_v$ is $p_v(u) = 1 - \prod_{w \in S \cap (H_1 \cup H_2 \cup \cdots \cup H_n)}(1 - p_{wu})$. Then, we have

$$\begin{aligned}
p_v(u) &= 1 - \prod_{w \in S \cap (H_1 \cup H_2 \cup \cdots \cup H_n)} (1 - p_{wu}) \\
&\geq 1 - \prod_{w \in \bigcup_{H_i \in S} H_i} (1 - p_{wu}) \\
&= 1 - \prod_{H_i \in S} \prod_{w \in H_i} (1 - p_{wu}) \\
&= 1 - \prod_{H_i \in S} \prod_{w \in H_i} \left(1 - \left(1 - (1 - p_{H_i,u})^{\frac{1}{|H_i|}}\right)\right) \\
&= 1 - \prod_{H_i \in S} \prod_{w \in H_i} \left(1 - p_{H_i,u})^{\frac{1}{|H_i|}}\right)^{|H_i|} \\
&= 1 - \prod_{H_i \in S} (1 - p_{H_i,u}) \\
&= p(u).
\end{aligned}$$

Therefore, for each inactivated node u, $p_v(u) \geq p(u)$, we can know that $S_1 \subseteq S_1^v$ after the first round.

When $t \neq 0$, we need to prove $S_t \subseteq S_t^v \Rightarrow S_{t+1} \subseteq S_{t+1}^v$. Then, for each inactivated node u, the probability that u can be activated in $\bar{G}$ is $p(u) = 1 - \prod_{H_i \subseteq S_t}(1 - p_{H_i,u})$, where there exists n hyperedges $\{(H_1, u), (H_2, u), \ldots, (H_n, u)\}$ in $\bar{G}$. The probability that u can be activated in $\bar{G}_v$ is $p_v(u) = 1 - \prod_{w \in S_t^v \cap (H_1 \cup H_2 \cup \cdots \cup H_n)}(1 - p_{wu})$. Then, we have

$$\begin{aligned}
p_v(u) &= 1 - \prod_{w \in S_t^v \cap (H_1 \cup H_2 \cup \cdots \cup H_n)} (1 - p_{wu}) \\
&\geq 1 - \prod_{w \in S_t \cap (H_1 \cup H_2 \cup \cdots \cup H_n)} (1 - p_{wu}) \\
&\geq 1 - \prod_{w \in \bigcup_{H_i \in S_t} H_i} (1 - p_{wu}) \\
&= 1 - \prod_{H_i \in S_t} \prod_{w \in H_i} (1 - p_{wu}) \\
&= 1 - \prod_{H_i \in S_t} \prod_{w \in H_i} \left(1 - \left(1 - (1 - p_{H_i,u})^{\frac{1}{|H_i|}}\right)\right) \\
&= 1 - \prod_{H_i \in S_t} \prod_{w \in H_i} \left(1 - p_{H_i,u})^{\frac{1}{|H_i|}}\right)^{|H_i|} \\
&= 1 - \prod_{H_i \in S_t} (1 - p_{H_i,u}) \\
&= p(u)
\end{aligned}$$

Therefore, for each inactivated node u, $p_v(u) \geq p(u)$, we can know that $S_{t+1} \subseteq S_{t+1}^v$ after $t + 1$ round. ■

B. Lower Bound

The main idea in [26] to get a lower bound is to keep only such hyperedge whose nodes in head set have the same head set, otherwise delete it. Suppose that there are a head set W such that for each $u \in H_e$, (W, u) exists in $\bar{G}$, this hyperedge (H_e, v) is reserved. Otherwise, we delete it. In addition, H_e and W can be replaced by super node, and the new directed edges (w, h) and (h, v) should be added into graph. The weight of super node will be 0 and other common nodes will be 1. Therefore, the problem transform into IM with weighted case [26]. This method is suitable for us to construct directed graph $\bar{G}_\mu = (V, \bar{E}_\mu)$ for lower bound. Because we deleted these hyperedges whose head set cannot be activated simultaneously, so the eventual expected influence will be reduced. Then, the lower bound problem for IMCCP is formulated.

Definition 12 (Lower Bound Problem): Given the graph $\bar{G}_\mu = (V, \bar{E}_\mu)$, $c = \{c_1, c_2, \ldots, c_t\}$ and budget B, find the seed set S that satisfies budget constraint, $c(S) \leq B$, and maximize the expected spread $\mu(S)$.

Here, we transform hypergraph $\bar{G} = (V, \bar{E})$ to directed graph $\bar{G}_\mu = (V, \bar{E}_\mu)$, which is equivalent to the constructed graph in IMCP problem; thus, it is nondecreasing and submodular. We can use Algorithms 2 and 3 to solve it and obtain a $(1/2)(1 - (1/e))$ approximation ratio.

Theorem 13: Given $\bar{G}_\mu = (V, \bar{E}_\mu)$, $\mu(S)$ is a lower bound of $\sigma(S)$ with respect to seed set S.

Proof: $\mu(S) \leq \sigma(S)$ for $S \subseteq V$ if $\mu(S)$ is an lower bound of $\sigma(S)$. We define that S_t and S_t^μ are the activated node in $\bar{G}$ and $\bar{G}_\mu$ at the time step t. At any time step, S_t^μ should be the subset of S_t. The proof is very similar to [26], thus we will not go into details here. ■

VI. Sandwich Framework

Sandwich approximation strategy was first proposed by Lu *et al.* [6], which is an algorithm with data-dependent approximation factor when the objective function is not submodular. The objective function $\sigma(\cdot)$ is not submodualr, we define $\mu(\cdot)$ and $\nu(\cdot)$ on the same ground set V such that $\mu(S) \leq \sigma(S) \leq \nu(S)$ for $S \subseteq V$. If $\mu(\cdot)$ and $\nu(\cdot)$ are submodular function under the knapsack constraint, they can be approximated within $(1/2)(1 - (1/e))$ by Algorithms 2 and 3. Then, we can sandwich $\sigma(\cdot)$ with $\mu(\cdot)$ and $\nu(\cdot)$ to maximize $\sigma(\cdot)$ as follows:

$$S_{\text{sand}} = \arg \max_{S \in \{S_\mu, S_\sigma, S_\nu\}} \sigma(S) \tag{5}$$

where S_μ, S_σ, and S_ν be the solution from $\mu(\cdot)$, $\sigma(\cdot)$, and $\nu(\cdot)$. From Section V, we obtain the submodular upper bound $\nu(\cdot)$ and lower bound $\mu(\cdot)$ for $\sigma(\cdot)$. Then, the sandwich approximation framework is shown in Algorithm 4.

Theorem 14: Let S_{sand} be the seed set returned by Sandwich approximation framework, then we have

$$\sigma(S_{\text{sand}}) \geq \max\left\{\frac{\sigma(S_\nu)}{\nu(S_\nu)}, \frac{\mu(S^*)}{\sigma(S^*)}\right\} \cdot \frac{1}{2}\left(1 - \frac{1}{e}\right) \cdot \sigma(S^*) \tag{6}$$

where S^* is the optimal solution for maximizing $\sigma(S)$ subject to budget $c(S^*) \leq B$.

Algorithm 4 Sandwich Approximation Framework

Input: Graph $G = (V, E)$, budget B and costs c
Output: Seed set S_{sand}
1: Initialize upper bound graph $\bar{G}_\nu = (V, \bar{E}_\nu)$
2: Initialize lower bound graph $\bar{G}_\mu = (V, \bar{E}_\mu)$
3: $S_\nu = \arg\max \nu(S)$ in $\bar{G}_\nu = (V, \bar{E}_\nu)$
4: $S = \arg\max \sigma(S)$ in $\bar{G} = (V, \bar{E})$
5: $S_\mu = \arg\max \mu(S)$ in $\bar{G}_\mu = (V, \bar{E}_\mu)$
6: $S_{sand} = \arg\max_{S_0 \in \{S_\nu, S, S_\mu\}} \sigma(S_0)$ by Algorithm 2 or 4
7: **return** S_{sand}

Proof: Let S_μ^* and S_ν^* be the optimal solution to maximizing $\mu(S)$ and $\nu(S)$ subject to budget $c(S_\mu^*) \leq B$ and $c(S_\nu^*) \leq B$, respectively. We have

$$\begin{aligned}\sigma(S_\nu) &= \frac{\sigma(S_\nu)}{\nu(S_\nu)} \cdot \nu(S_\nu) \\ &\geq \frac{\sigma(S_\nu)}{\nu(S_\nu)} \cdot \frac{1}{2}\left(1 - \frac{1}{e}\right) \cdot \nu\left(S_\nu^*\right) \\ &\geq \frac{\sigma(S_\nu)}{\nu(S_\nu)} \cdot \frac{1}{2}\left(1 - \frac{1}{e}\right) \cdot \nu(S^*) \\ &\geq \frac{\sigma(S_\nu)}{\nu(S_\nu)} \cdot \frac{1}{2}\left(1 - \frac{1}{e}\right) \cdot \sigma\left(S_\nu^*\right).\end{aligned}$$

Then, observed slightly from lower bound, we can get that as follows:

$$\begin{aligned}\sigma(S_\mu) &\geq \mu(S_\mu) \\ &\geq \frac{1}{2}\left(1 - \frac{1}{e}\right) \cdot \mu\left(S_\mu^*\right) \\ &\geq \frac{1}{2}\left(1 - \frac{1}{e}\right) \cdot \mu(S^*) \\ &\geq \frac{\mu(S^*)}{\sigma(S^*)} \cdot \frac{1}{2}\left(1 - \frac{1}{e}\right) \cdot \sigma(S^*).\end{aligned}$$

Therefore,

$$\begin{aligned}\sigma(S_{\text{sand}}) &= \max\{\sigma(S_\nu), \sigma(S), \sigma(S_\mu)\} \\ &\geq \max\{\sigma(S_\nu), \sigma(S_\mu)\} \\ &= \max\left\{\frac{\sigma(S_\nu)}{\nu(S_\nu)}, \frac{\mu(S^*)}{\sigma(S^*)}\right\} \cdot \frac{1}{2}\left(1 - \frac{1}{e}\right) \cdot \sigma(S^*).\end{aligned}$$

The theorem is proven. ■

VII. Experiment

In this section, we show the effectiveness and efficiency of our proposed algorithms on two real social networks. Our goal is to evaluate Algorithms 2 and 3 with some commonly used baseline algorithms. In addition, we evaluate sandwich approximation framework on predefined hypergraphs.

A. Data Set Description and Statistics

Our experiments are based on the data set from networkrepository.com, which is an online network repository containing different kinds of network. There are two data sets in our experiments. The data set-1 is a coauthorship network [27], namely, coauthorship of scientists in network theory and experiments. The data set-2 is a Wiki-vote network [27], namely, Wikipedia who-votes-on-whom network. They are classified as social networks in networkrepository.com. Basic statistics of these data sets are summarized in Table I, where n denotes the number of nodes and m denotes the number of edges in the original graph. However, the real networks are different from these basic information, which are related to the number of products, we will describe in detail later.

TABLE I
Statistics of Two Data Sets

Dataset	n	m	Type	Average degree
dataset-1	379	914	directed	4
dataset-2	914	2914	directed	6

B. Experimental Setup

Two experiments are performed for each data set. The first experiment is performed to obtain expected influence $\sigma(S)$ for a given budget B (range from 0 to 40) for IMCP. The second experiment is to solve IMCCP by use of sandwich approximation framework. These experiments are based on IC-model, so we need to set the propagation probability for each edge and the cost for each product. In the first experiment, it can be divided into three subcases as follows.

1) Assuming there are two products A and B, the constructed graph $G = (V, E)$ has two layers, one is product A and the other is product B. The propagation probability for A is $p_A = 0.1$, B is $p_B = 0.12$, $p_{A\to B} = 0.11$, and $p_{B\to A} = 0.11$. The cost is $c_A = 1$, $c_B = 1.2$. Therefore, the total nodes and edges in data set-1 network are 758 and 2586, respectively, and in data set-2 network is 1778 and 7606.
2) Assuming there are three products A, B, and C, the constructed graph has three layers. Except A and B, the propagation probability for C is $p_C = 0.14$ and $p_{A\to C} = 0.11$. The cost is $c_C = 1.3$.
3) Assuming there are four products A, B, C, and D, their constructed graph has four layers. Except A, B, and C, the propagation probability for D is $p_D = 0.15$ and $p_{D\to C} = 0.11$. The cost is $c_D = 1.5$. We compare both of our algorithms (Greedy and general-TIM) with some common baseline algorithms. The algorithm Random selected the seeds randomly with the restraint of budget from the constructed graph, and max-degree selects nodes with the highest out-degree as the seeds.

In the second experiment, it can be divided into two subcases as follows.

1) Assuming there are three products A, B, and C, so the constructed graph $G = (V, E)$ has three layers, one is product A, one is for product B, and the other is product C. The propagation probability for A is $p_A = 0.12$, B is $p_B = 0.1$, $p_C = 0.1$, and $p_{BC\to A} = 0.2$. Here, hyperedge $(\{u_B, u_C\}, u_A)$ exists in the constructed graph. The cost is $c_A = 1.2$, $c_B = 1$, and $c_C = 1$.

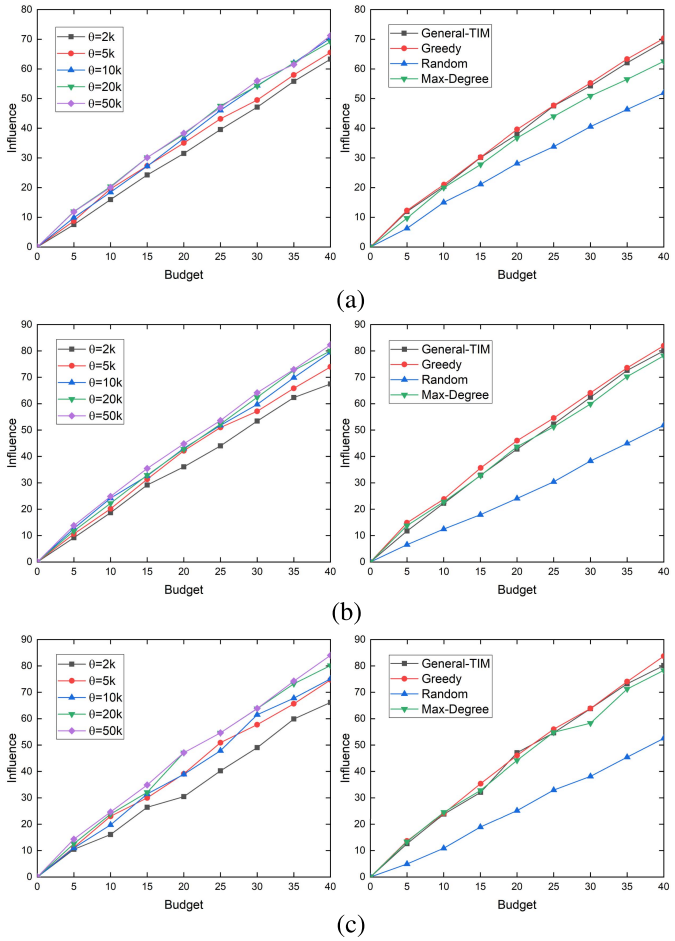

Fig. 3. Performance comparison achieved by different algorithms with budget 40 under the data set-1 in the first experiment. Here, the left column is the expected influenced achieved by general-TIM with different θ. (a) Subcase: two products. (b) Subcase: three products. (c) Subcase: four products.

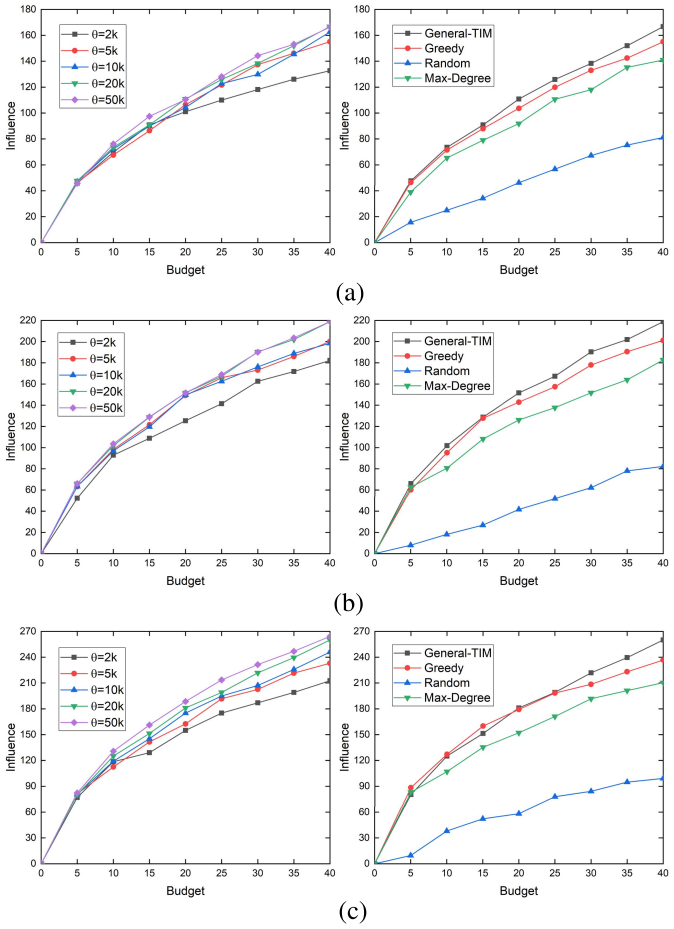

Fig. 4. Performance comparison achieved by different algorithms with budget 40 in the first experiment. Here, the left column is the expected influenced achieved by general-TIM with different θ. (a) Subcase: two products. (b) Subcase: three products. (c) Subcase: four products.

In addition, in upper bound constructed graph $\bar{G}_v = (V, \bar{E}_v)$, we replace $(\{u_B, u_C\}, u_A)$ with (u_B, u_A) and (u_C, u_A), here $p_{B\to A} = 0.1056$ and $p_{B\to C} = 0.1056$. Then, in lower bound constructed graph $\bar{G}_\mu = (V, \bar{E}_\mu)$, we remove hyperedge $(\{u_B, u_C\}, u_A)$ directly.

2) Assuming there are four products A, B, C, and D, Except A, B, and C, shown as subcase 1), the propagation probability for D is $p_D = 0.13$ and $p_{A\to D} = 0.11$. The cost is $c_D = 1$. We use sandwich approximation framework to get approximation solutions, then compare with their upper bound and lower bound.

TABLE II
TIME CONSUMING IN THE FIRST EXPERIMENT

Dataset-1				
	Greedy	**General-TIM**	**Max-Degree**	**Random**
(a)	286.12	21.82	0.64	0.35
(b)	466.85	32.97	0.87	0.36
(c)	564.40	42.50	0.80	0.35
Dataset-2				
	Greedy	**General-TIM**	**Max-Degree**	**Random**
(a)	2356.07	48.08	2.57	0.91
(b)	4685.78	79.84	3.37	0.89
(c)	6602.19	99.95	3.75	1.34

C. Experimental Results

Figs. 3 and 4 draw the performance comparison achieved by different algorithms with budget 40 under data set-1 and data set-2 in the first experiment. Fig. 5 draws the expected influence achieved by the sandwich approximation framework with budges 40 in the second experiment.

Obviously, from Figs. 3 and 4, $\theta = 20\ k$ is large enough to make sure the solution is a good estimation, Because the difference between $\theta = 20\ k$ and $\theta = 50\ k$ is extremely small in two networks. In addition, the expected influence of the Greedy algorithm and general-TIM algorithm is very close in two networks under the different setting, which prove the effectiveness of the randomized sampling by RR-sets under the knapsack constraint. However, the general-TIM is much faster than Greedy, so it is more suitable in large networks. The time-consuming in the first experiment is shown in Table II, where the θ value of general-TIM is 20 k. Both the Greedy and general-TIM algorithms outperform other technique, which prove their effectiveness. Comparing with

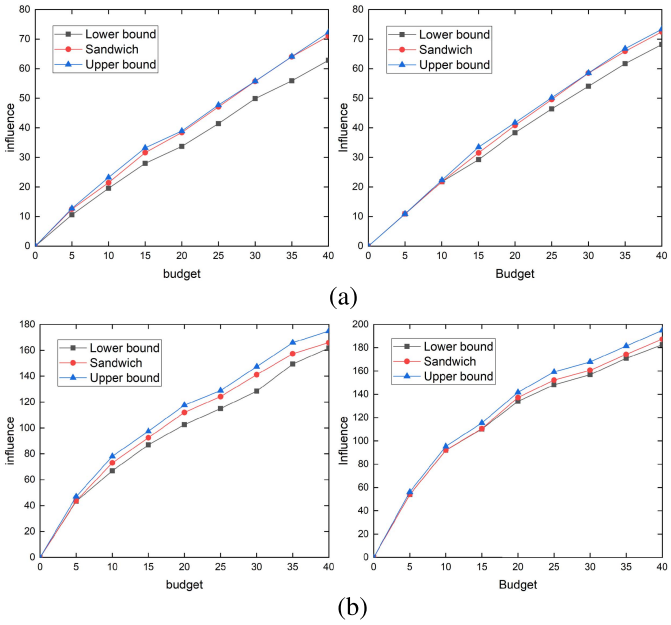

Fig. 5. Expected influence achieved by the sandwich approximation framework with budget 40 in the second experiment. (a) Subcase: two products. (b) Subcase: three products.

max-degree algorithm, which is the best baseline algorithm, Greedy and general-TIM algorithms are 20% better than it. From the above figures, we can see that comparing with the results of data set-1, the results of Greedy and general-TIM algorithms in data set-2 are better than baseline algorithm more clearly. This may be related to its network structure.

Although existing Com-IC model [6] solves influence diffusion from competition to complementarity. From above experiments and theoretical analysis before, we can see that our work is very different from Com-IC, mainly includes the following aspects: 1) we use multi-layer network structure, which is more flexible and tractable; 2) com-IC is only suitable to solve two products, and it cannot be extended to deal with multiple products because their model is based on original network; 3) the objective function of Com-IC is not submodular, which increases the complexity of the model; and 4) com-IC is impossible to solve the composite complementary problem. Thus, our model is better than Com-IC.

From Fig. 5, it is observed that the expected influence $\sigma(S)$ of sandwich approximation framework lies in between its upper bound and lower bound for the two networks. In addition, we can see that the upper bound is very close to $\sigma(S)$, which is better than the method in [26].

VIII. Conclusion

In this paper, we modeled the VMCP by using multi-layer constructed networks. IMCP was formulated to select initial users to adopt different products to maximize the expected influence with knapsack constraint under the IC-model. We proved that IMCP is NP-hard and submodular. In order to get better scalability, we put forward the general-TIM algorithm with the help of random RR-set. Considering CCP existing, IMCCP was formulated to adapt to CCP. We proved that IMCCP is NP-hard, nonsubmodular, and nonsupermodular. We use sandwich approximation framework to solve it because upper bound and lower bound can be acquired and obtained an approximation ratio. Finally, we tested our algorithm on two real-world data sets. The experimental result verified the effectiveness of general-TIM algorithm and sandwich approximation framework.

In future research, our research can be divided into two parts. The first part is to get better randomized sampling methods to solve submodular problem under the knapsack problem or prove a theoretical lower bound to θ. The second part is to solve nonsubmodular optimization (containing hyperedges) better by different methods.

References

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

[2] M. Richardson and P. Domingos, "Mining knowledge-sharing sites for viral marketing," in *Proc. 8th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2002, pp. 61–70.

[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*, Aug. 2003, pp. 137–146.

[4] W. Chen *et al.*, "Influence maximization in social networks when negative opinions may emerge and propagate," in *Proc. SIAM Int. Conf. Data Mining*, Apr. 2011, pp. 379–390.

[5] H. Zhang, S. Mishra, M. T. Thai, J. Wu, and Y. Wang, "Recent advances in information diffusion and influence maximization in complex social networks," *Opportunistic Mobile Social Netw.*, vol. 37, no. 1, p. 37, Aug. 2014.

[6] 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, Oct. 2015.

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

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

[9] W. Chen, L. V. S. Lakshmanan, and C. Castillo, "Information and influence propagation in social networks," *Synthesis Lectures Data Manage.*, vol. 5, no. 4, pp. 1–177, Oct. 2013.

[10] 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*, Aug. 2007, pp. 420–429.

[11] A. Goyal, W. Lu, and L. V. Lakshmanan, "Celf++: Optimizing the greedy algorithm for influence maximization in social networks," in *Proc. 20th Int. Conf. Companion World Wide Web*, Mar. 2011, pp. 47–48.

[12] 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.

[13] 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.

[14] 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.

[15] H.-C. Ou, C.-K. Chou, and M.-S. Chen, "Influence maximization for complementary goods: Why parties fail to cooperate?" in *Proc. 25th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2016, pp. 1713–1722.

[16] U. Feige and R. Izsak, "Welfare maximization and the supermodular degree," in *Proc. 4th Conf. Innov. Theor. Comput. Sci.*, Jan. 2013, pp. 247–256.

[17] M. Feldman and R. Izsak, "Constrained monotone function maximization and the supermodular degree," 2014, *arXiv:1407.6328*. [Online]. Available: https://arxiv.org/abs/1407.6328

[18] M. Feldman, "Building a good team: Secretary problems and the supermodular degree," in *Proc. 28th Annu. ACM-SIAM Symp. Discrete Algorithms*, 2017, pp. 1651–1670.
[19] W. Chen, T. Lin, Z. Tan, M. Zhao, and X. Zhou, "Robust influence maximization," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 795–804.
[20] Z. Wang, Y. Yang, J. Pei, L. Chu, and E. Chen, "Activity maximization by effective information diffusion in social networks," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 11, pp. 2374–2387, Nov. 2017.
[21] M. Narasimhan and J. A. Bilmes, "A submodular-supermodular procedure with applications to discriminative structure learning," 2012, *arXiv:1207.1404*. [Online]. Available: https://arxiv.org/abs/1207.1404
[22] R. Iyer and J. Bilmes, "Algorithms for approximate minimization of the difference between submodular functions, with applications," 2012, *arXiv:1207.0560*. [Online]. Available: https://arxiv.org/abs/1207.0560
[23] W.-L. Wu, Z. Zhang, D.-Z. Du, "Set function optimization," *J. Oper. Res. Soc. China*, vol. 7, no. 2, pp. 183–193, Jun. 2019. doi: 10.1007/s40305-018-0233-3.
[24] S. Khuller, A. Moss, and J. S. Naor, "The budgeted maximum coverage problem," *Inf. Process. Lett.*, vol. 70, no. 1, pp. 39–45, Apr. 1999.
[25] D.-Z. Du, K.-I. Ko, and X. Hu, *Design and Analysis of Approximation Algorithms*, vol. 62. New York, NY, USA: Springer-Verlag, 2011.
[26] J. Zhu, J. Zhu, S. Ghosh, W. Wu, and J. Yuan, "Social influence maximization in hypergraph in social networks," *IEEE Trans. Netw. Sci. Eng.*, to be published.
[27] R. A. Rossi and N. K. Ahmed, "The network data repository with interactive graph analytics and visualization," in *Proc. 29th AAAI Conf. Artif. Intell.*, Mar. 2015. [Online]. Available: http://networkrepository.com/soc.php

Jianxiong Guo received the B.S. degree in energy engineering and automation from the South China University of Technology, Guangzhou, China, in 2015, and the M.S. degree in chemical engineering from the University of Pittsburgh, Pittsburgh, PA, USA, in 2016. He is currently pursuing the Ph.D. degree with the Department of Computer Science, University of Texas at Dallas, Richardson, TX, USA.

His current research interests include social networks, data mining, and design of approximation algorithm.

Weili Wu (M'00) received the Ph.D. and M.S. degrees from the Department of Computer Science, University of Minnesota, Minneapolis, MN, USA, in 2002 and 1998, respectively.

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