Profit Maximization for Competitive Influence Spread in Social Networks

Qiufen Ni[1], Yun Bai[1], and Zhongzheng Tang[2](✉)

[1] School of Computer Science and Technology, Guangdong University of Technology,
Guangzhou 510006, China
niqiufen@gdut.edu.cn

[2] School of Science, Beijing University of Posts and Telecommunications,
Beijing 100876, China
tangzhongzheng@amss.ac.cn

Abstract. Influence maximization is a classic problem in social networks and has been extensively studied in recent years. Viral marketing is an important application for influence maximization. Most of existing related research focus on influence maximization of a single product, but in reality, a marketer may promote multiple products in the social network at the same time. This paper studies the profit maximization problem for multiple kinds of products in viral marketing. We formulate it as the Profit Maximization Problem for Competitive Influence Spread (PMPCIS), which aims at selecting a set of seed users within the total budget B and the total number of seeds K to maximize the overall profit of k kinds of products. The objective problem is proved to be a monotone k-submodular maximization problem under the knapsack and cardinality constraint. We present a Singleton+Greedy-Local-Search Algorithm in four steps, and prove the approximation performance guarantee of the proposed algorithm.

Keywords: Social Network · Profit Maximization · k-submodular

1 Introduction

Online social networks, such as WeChat, Facebook, Twitter, have been important platforms for people to communicate and for business to advertise. People keep in touch with each other and make friends through social networks, they also like to share their innovations and ideas, etc., in the social networks [1,2]. According to statistics, there are 3.725 billion users active in the social networks by Dec. 2019. Companies make use of the advantage of large crowds and rapid information dissemination in social networks to promote their products. Motivated by the information propagation in social networks, Influence Maximization (IM) problem is put forward by Kempe *et al.* in [3]. They formulate the IM problem as: selecting a set of users as seeds to maximize the expected number of users who are influenced by seeds. Influence maximization finds applications in many domain, like viral marketing. Viral marketing makes good use

W. Wu and G. Tong (Eds.): COCOON 2023, LNCS 14423, pp. 366–377, 2024.
https://doi.org/10.1007/978-3-031-49193-1_28

of the word-of-mouth effect of the social networks, and it promotes products by giving discounts to a small set of customers to spread their product information. Two classic influence spread model: Independent Cascade (IC) model and Linear Threshold (LT) model, are proposed by Kempe *et al.* in [3]. They also prove that the influence propagation function is monotone and submodular, thus, the greedy algorithm can be applied to solve the IM problem and obtain an $1 - 1/e - \epsilon$ approximation ratio.

In real life, it is very common for multiple types of information propagate simultaneously in the social network. For example, in 2022 Apple Corporation issues two kinds of new iPhone, iPhone 14 is more cost-effective and iPhone 14 Pro is more powerful but more expensive. Although iPhone 14 and iPhone 14 Pro are the same kind of product, but they have different prices, appearances and performances, which can attract different groups of customers with different requirements. Based on the described background, consider the scenario where there are k kinds of products, a constraint K for the number of selected seeds, and a budget B for the activation cost of all selected seeds. Assuming that different seeds have different activation costs and different profits can be obtained when the product is purchased, which nodes to be selected as seeds and how to allocate the budget to seeds for k kinds of products such that the total profit is maximized? We study the profit maximization problem for competitive products in this paper, and assume that each seed can accept the discount of only one product for the fairness. We aim to allocate discounts to k sets of seed users for k kinds of products under two constraints K and B. The objective function that maximizes the total profit of k kinds of products can be formulated as a k-submodular function. Approximation algorithms with theoretical guarantee are proposed in our work. We summarize the main contributions in this paper as follows:

- We formulate the profit maximization for competitive influence spread problem as a k-submodular function with both a knapsack constraint and a cardinality constraint problem. To best of our knowledge, this is the first time to study a monotone k-submodular maximization problem under both the knapsack and cardinality constraints in social networks.
- We propose a Singleton+Greedy-Local-Search Algorithm in four steps, which obtains two approximation ratios: 0.216 and 0.158, in two different conditions.

2 Related Work

In this paper, we consider the profit maximization problem with multiple kinds of products in the network. We summarize the related studies on our work as follows.

Influence Maximization with Competitive Influence Spread: Most of the existing relevant studies on influence maximization consider the scenario of a single kind of information spreading in the social network. The competitive IM problem is firstly studied by Bharathi *et al.* [4]. They propose a game theory based method to solve it. Liang *et al.* [5] consider that multiple kinds of similar products promote in the network at the same time and the target of the promotion is a specific user group. They formulate such a problem as Targeted Influence Maximization in Competitive social networks (TIMC). A reverse reachable set based greedy method is proposed by them to solve the TIMC with approximation performance guarantee. Wu *et al.* [6] consider a scenario that multiple information propagate in a social network with different propagation probabilities. The problem is formulated as maximizing the total influence of all the different information under a constraint of seed budget k. They present a greedy algorithm with $\frac{1}{3}$ approximation ratio, and also propose a parallel algorithm which improves the efficiency of the algorithm.

Profit Maximization: Profit maximization problem is a transformation of the influence maximization problem, which aims at maximizing the profit of products by selecting a seed set with a limited budget. Zhang *et al.* [7] study the Profit Maximization with Multiple Adoptions (PM^2A) problem. Two different approximation algorithms are devised to maximize the total profit of multiple kinds of product by selecting a limited number of seeds. Chen *et al.* [8] propose a Randomized Modified Greedy (RMG) algorithm to solve the Profit Maximization with Multiple Adoptions (PM^2A) problem, which obtain an $(1 - 1/e - \epsilon)$ approximation ratio. Yuan *et al.* [9] design two discount allocation strategies under the non-adaptive setting and adaptive setting, respectively, which achieve the goal of maximizing the expected number of users who adopt the product finally.

k-Submodular Maximization: Huber *et al.* [10] firstly define the k-submodular function, which is a generalization of the submodular function. Ohsaka *et al.* [11] study the maximization for a monotone k-submodular function with two different size constraints, and propose greedy algorithms with constant approximation factor. Tang *et al.* [12] study the maximization for a non-negative monotone k-submodular function with a knapsack constraint, they present a greedy algorithm which can obtain an $\frac{1}{2} - \frac{1}{2e}$ approximation ratio with an $O(n^4k^3)$ time complexity. Wang *et al.* [13] propose a framework for relaxing a k-submodular function to continuous space with the technique of multilinear extension. They also improve the approximation ratio to $1/2 - \epsilon$ for maximizing a monotone k-submodular function with knapsack constraint. V. Pham *et al.* [14] explore the applications of maximizing the k-submodular function under the knapsack constraint in influence maximization of social networks and the sensor placement. However, for the monotone k-submodular maximization problem under the knapsack constraint and cardinality constraint, to best of our knowledge, there is no related conclusion about it in social networks. Thus, we try to fill this gap and apply it to the social networks.

3 Diffusion Model and Problem Definition

3.1 The Diffusion Model

A social network is constructed as a directed graph $G(V, E)$, where each node $v \in V$ represents a user, and each edge $(u, v) \in E$ represents that user v follows user u, u is an incoming neighbor of v, while v is an outgoing neighbor of u. The incoming neighbor set and the outgoing neighbor set of a node v are denoted as $N^-(v)$ and $N^+(v)$, respectively. IC model is used as the influence propagation model in our problem, its influence propagation process is described as follows.

Definition 1 (IC model). *Nodes in social networks have two different states: active and inactive, all nodes are initially inactive. There is a activation probability $p_{uv} \in (0, 1]$ associated with each edge $e = (u, v) \in E$. When a node u is firstly activated at time t, then for each of his inactive outgoing neighbor $v \in N^+(u)$, he can activate them with a probability p_{uv} at time $t+1$. Finally, the influence propagation process terminates if there are no newly activated nodes in the future.*

3.2 Problem Formulation

Give a social network $G = (V, E)$. Assume that marketers wants to promote k kinds of products in the social network. k kinds of products information propagate under the IC model at the same time. We aim to choose k seed sets $S = \{S_1, S_2, \cdots, S_k\}$ and provide discounts to them, and assume that each seed user can only be used for propagating at most one kind of product information. As different user can bring different levels of influence, so we give different discounts to different seeds, and influential users get bigger discounts. Let $\sigma(S_i)$ be the expected influence spread of seed set S_i for product i, i.e., the expected number of users who adopt product i in the social network. Let $f(S_i)$ be the total profit that obtained by purchasing product i. Moreover, $\sigma(S|G)$ and $f(S|G)$ are the expected number of influenced people and the total profit obtained by adopting k kinds of products, respectively.

The profit maximization problem for competitive products marketing at the same time in the social network with a total activation cost B and a total number of seeds K constraints can be formulated as follows:

Problem 1 ***(Profit Maximization Problem for Competitive Influence Spread (PMPCIS)).*** *Given a social network graph $G = (V, E)$, k kinds of products, the IC model, the cost $c(a)$ that activating a node a to purchase a product, the profit p_i that a node can gain when he adopts product i, the seed set $S = \{S_1, S_2, \cdots, S_k\}$, where S_i is selected for propagating the information of the product i and $S_i \cap S_j = \varnothing$ for any $i, j \in [1, k]$ and $i \neq j$. The total number of selected seeds is represented by $|S| = \sum_{i=1}^{k} |S_i|$ and the upper bound is K. The total activation cost for the seed set S is denoted by $c(S) = \sum_{i=1}^{k} \sum_{a \in S_i} c(a)$ and the total budget is given as B. The expected influence spread for seed set S_i is expressed by $\sigma(S_i)$. Our target is to select an optimal seed set $S = \{S_1, S_2, \cdots, S_k\}$ such that the total profit $f(S)$ is maximized, i.e.,*

$$\begin{aligned} S^* = \arg\max_{S} \ & f(S) \\ s.t. \quad & c(S) \leq B \\ & |S| \leq K. \end{aligned} \tag{1}$$

From the definition of PMPCIS, we can intuitively get $f(S)=\sum_{i=1}^{k} p_i\sigma(S_i)$. In literature [3], Kempe *et al.* have proved that the classical influence maximization problem in IC model is a NP-hard problem. When the types of products are reduced to one, our PMPCIS is equivalent to the traditional IM problem for IC model. Thus, the PMPCIS is a NP-hard problem.

4 Solution for PMPCIS

We propose solution for PMPCIS in this section. At first, we analyze the properties of the objective function f for PMPCIS.

4.1 Properties for Objective Function f

Firstly, we introduce an important property for a set function: k-submodular. Let X be a finite non-empty set, and let $(k+1)^X := \{(U_1,\cdots,U_k) \mid U_i \subseteq X, \forall i \in \{1,2,\cdots,k\}, U_i \cap U_j = \varnothing, \forall i \neq j\}$ be the family of k disjoint sets $(U_1,\cdots,U_k)$. A function h: $(k+1)^X \to \mathbb{R}$ is k-submodular if for any $\boldsymbol{U}=\{U_1,\cdots,U_k\}$ and $\boldsymbol{W}=\{W_1,\cdots,W_k\}$ in $(k+1)^X$, it satisfies,

$$h(\boldsymbol{U})+h(\boldsymbol{W}) \geq h(\boldsymbol{U}\sqcup\boldsymbol{W})+h(\boldsymbol{U}\sqcap\boldsymbol{W}),$$

where

$$\boldsymbol{U}\sqcap\boldsymbol{W} := (U_1\cap W_1,\cdots,U_k\cap W_k),$$

$$\boldsymbol{U}\sqcup\boldsymbol{W} := \left(U_1\cup W_1\backslash\left(\bigcup\nolimits_{i\neq 1} U_i\cup W_i\right),\cdots,U_k\cup W_k\backslash\left(\bigcup\nolimits_{i\neq k} U_i\cup W_i\right)\right).$$

If a function satisfies the properties of orthant submodularity and pairwise monotonicity at the same time, which indicates that it is a k-submodular function. It is very intuitively to verify that our objective function is k-submodularity.

Then, we elaborate on additional characteristics and notations of k-submodular functions for our problem. Every k-tuple $\mathbf{x}=(X_1,\ldots,X_k)\in(k+1)^V$ uniquely corresponds to a set $A=\{(a,d)\mid a\in X_d, d\in[k]\}$ composed of *item-dimension pairs*. Hence, a user-product pair (a,d) is included in set A (termed as a *solution*) if and only if $a\in X_d$ in $\mathbf{x}$.

In our problem, for ease of presentation, we write $\mathbf{x}$ and its corresponding solution A interchangeably. For any solution $A\in(k+1)^V$, we define $U(A):=\{a\in V\mid \exists\ d\in[k],\ (a,d)\in A\}$ to be the set of seed included, and the *size* is $|A|=|U(A)|$.

The marginal gain of adding a user-product pair (a,d) to A is

$$\varDelta_{a,d}f(A)=f(A\cup\{(a,d)\})-f(A),$$

and the marginal density is $\frac{\varDelta_{a,d}f(A)}{c(a)}$. As the profit maximization function f is monotone k-submodular, it satisfies the *pairwise monotonicity*

$$\varDelta_{a,d}f(A)+\varDelta_{a,l}f(A)\geq 0,$$

for any $d, l \in [k]$ and $d \neq l$. And it also satisfies the *orthant submodularity*

$$\Delta_{a,d} f(A) \geq \Delta_{a,d} f(C),$$

$$\forall A, C \in (k+1)^V \text{ with } A \subseteq C, a \notin U(C), d \in [k].$$

In this problem, each seed node $a \in V$ has a non-negative cost $c(a)$, and the cost of solution A is $c(A) = \sum_{a \in U(A)} c(a)$. The goal is to find a solution that maximizes the function value within the given budget $B \in \mathbb{Z}_+$ and size $K \in \mathbb{Z}_+$. Then we rewrite the PMPCIS in Eq. (1) as follows.

$$\begin{aligned} A^* = \arg \max_{A \in (k+1)^V} & \ f(A) \\ s.t. \quad c(A) & \leq B \\ |A| & \leq K. \end{aligned} \tag{2}$$

4.2 Proposed Algorithm

We present our solution for the proposed PMPCIS in four steps. In the first step, we consider unconstrained profit maximization problem and propose a simple greedy algorithm in Algorithm 1. Given a node set $V' = \{e_1, e_2, \ldots, e_m\}$ and $V' \subseteq V$. Firstly, we remove the two constraints of the problem in Eq. 2, then, Algorithm 1 is devised by greedily finding the maximum function value $\max_{A \in (k+1)^{V'}} f(A)$ without any constraint.

Let $T = \{(e_1, d_1^*), \ldots, (e_m, d_m^*)\}$ be an optimal solution that maximizes the function $f(A)$ over V'. Assume without loss of generality that the seed set are obtained by **Greedy Algorithm** in the order of $\{e_1, e_2, \ldots, e_m\}$, and denote the returned greedy solution by $A = \{(e_1, d_1), \ldots, (e_m, d_m)\}$. For $j = \{0, 1, \ldots, m\}$, define $A_j = \{(e_1, d_1), \ldots, (e_j, d_j)\}$. We have $A_0 = \varnothing$ and $A_m = A$. The following Lemma 1 is crucial to our subsequent proof.

Lemma 1 **([15,16]).** *If f is monotone, for $t = \{0, 1, \ldots, m\}$, we have $f(T) \leq 2f(A_t) + \sum_{e_i \in U(T) \setminus U(A_t)} \Delta_{e_i, d_i^*} f(A_t)$.*

Algorithm 1. A Simple Greedy Algorithm

Input: Social network subgraph $G = (V', E')$, objective function f.
Output: A solution $A \in (k+1)^{V'}$.

```
$A \leftarrow \varnothing$
for each item $a \in V'$ do
    $d_a \leftarrow \arg\max_{d \in [k]} \Delta_{a,d} f(A)$
    $A \leftarrow A \cup \{(a, d_a)\}$
end for
return $A$
```

Algorithm 2. Greedy-Knapsack Algorithm

```
Input: Social network graph $G = (V, E)$, objective function $f$, costs $c(a)$ for $a \in V$,
    budget $B$, seed size constraint $K$.
Output: Solution $A_t$ in $(k+1)^V$.
$A_0 \leftarrow \varnothing$, $V_0 \leftarrow V$, card $\leftarrow 0$
for $t$ from 1 to $|V|$ do
    Let $(a_t, d_t) = \arg\max_{a \in V_{t-1}, d \in [k]} \frac{\Delta_{a,d} f(A_{t-1})}{c(a)}$ maximize the marginal density, and
    denote $\rho_t = \frac{\Delta_{a_t,d_t} f(A_{t-1})}{c(a_t)}$.
    if $c(A_{t-1}) + c(a_t) \leq B$ then
        $A_t \leftarrow A_{t-1} \cup \{(a_t, d_t)\}$
        card $\leftarrow$ card$+1$
        if card $=$ $K$ then
            break
        end if
    else
        $A_t \leftarrow A_{t-1}$
    end if
    $V_t \leftarrow V_{t-1} \backslash \{a_t\}$
end for
return $A_t$.
```

Then we propose the following Algorithm 2. In Algorithm 2, we select the user who can bring the largest marginal density for objective function f in each iteration, and the selected users can not violate the budget and size constraint at the same time. Algorithm 2 returns the solution A_t.

Next, we devise the Algorithm 3, which aims at further optimizing the solution A_t. If we have solution with size K that is obtained with the Algorithm 2, we can execute the local search procedure utilizing Algorithm 3. We input the solution obtained by Algorithm 2, and denote it as a feasible solution A'. In each iteration, we try to swap a pair of selected and unselected items, aiming to augment the objective value while ensuring adherence to the knapsack constraint. In Algorithm 3, when we obtain the new seed set after the exchange in step 6, we invoke the Algorithm 1 **Greedy Algorithm** in step 7 to reassign the type of product that the seeds correspond to since the selected seed set may change.

Our main algorithm is as shown in Algorithm 4, which combines the singleton optimum, the greedy strategy in Algorithm 2 and the local search strategy in Algorithm 3. In step 1 of Algorithm 4, we find out the node that can maximize the objective function f among all the nodes, then A^* is the item pair with a single node set and the product it corresponds.

Algorithm 3. Local-Search Algorithm

```
Input: Social network graph $G = (V, E)$, objective function $f$, costs $c(a)$ for $a \in V$, budget $B$, seed size constraint $K$. a feasible solution $A' \in (k+1)^V$ with $|A'| = K$.
Output: A local optimal solution $A_t \in (k+1)^V$.
$A_K \leftarrow A'$, $t = K$, swap $\leftarrow$ true
while swap do
    swap $\leftarrow$ false
    Let $\rho_{(x,y)} = \max_{d \in [k]} \frac{f(A_t \backslash (y, d_y) \cup (x, d)) - f(A_t)}{c(x)}$ be the exchange marginal density. The swap $(a_{t+1}, b_{t+1})$ achieves the value $\max_{x \in V \backslash U(A_t), (y, d_y) \in A_t} \rho(x, y)$ and $\rho_{t+1} = \rho(a_{t+1}, b_{t+1})$
    if $\rho_{t+1} > 0$ and $c(a_{t+1}) - c(b_{t+1}) + c(A_t) \leq B$ then
        $V' \leftarrow U(A_t) \backslash \{b_{t+1}\} \cup \{a_{t+1}\}$
        $A_{t+1} \leftarrow$ Greedy$(V', f)$
        swap $\leftarrow$ true
        $t \leftarrow t + 1$
    end if
end while
return $A_t$
```

Algorithm 4. Singleton+Greedy-Local-Search Algorithm

```
Input: Social network graph $G = (V, E)$, objective function $f$, costs $c(a)$ for $a \in V$, budget $B$, seed size constraint $K$.
Output: The solution $A^*$ in $(k+1)^V$.
 1: Let $A^* \in \arg \max_{A:\ |A|=1, c(A) \leq B} f(A)$ be a size-1 solution giving the largest value.
 2: $A \leftarrow$ Greedy-Knapsack$(V, f, c, B, K)$
 3: if $f(A) > f(A^*)$ then
 4:     $A^* \leftarrow A$
 5: end if
 6: if $|A| < K$ then
 7:     return $A^*$
 8: end if
 9: $A \leftarrow$ Local-Search$(V, f, c, B, K, A)$
10: if $f(A) > f(A^*)$ then
11:     $A^* \leftarrow A$
12: end if
13: return $A^*$
```

If $f(A^*) \geq f(A)$, then A^* is returned as the final solution. We always find the node that maximize the marginal density as seeds in Algorithm 3, which may miss some nodes that can bring large profit and also need large costs at the same time, but have small marginal density. However, such nodes are actually a good solution. So the step 1 in Algorithm 4 is to find such kind of nodes. The returned solution A^* by the Algorithm 4 is a seed-product pair set, but in our original objective function in Eq. 1, we need to find an optimal seed set S^*, so the nodes in A^* is the returned solution for our PMPCIS in Eq. (1).

4.3 Approximation Performance Analysis

In this section, we analyze the approximation performance guarantee of the proposed Algorithm 4. We intend to establish the approximation ratio in Theorem 1 following the proof framework in [17]. The subsequent observation will be used twice in the proof of Theorem 1.

Lemma 2. *Let $A^1, A^2 \in (k+1)^V$ be two node sets with $|A^1| \leq K$ and $|A^2| = K$, then for any one-to-one function $y : U(A^1)\backslash U(A^2) \to U(A^2)\backslash U(A^1)$, we have $\sum_{x\in U(A^1)\backslash U(A^2)} f(A^2) - f(A^2\backslash\{(y(x), d_{y(x)})\}) \leq f(A^2)$ where $(y(x), d_{y(x)}) \in A^2$.*

Proof. Let $U(A^1)\backslash U(A^2) = \{x_1, x_2, \dots, x_{K'}\}$. For any one-to-one function y, suppose $y_j = y(x_j)$ for each $j \in \{1, \dots, K'\}$. Then we have

$$\begin{aligned}
&\sum_{x\in U(A^1)\backslash U(A^2)} f(A^2) - f(A^2\backslash\{(y(x), d_{y(x)})\}) \\
&\leq \sum_{j=1}^{K'} f(A^2\backslash\{(y_1, d_{y_1}), \dots, (y_{j-1}, d_{y_{j-1}})\}) - f(A^2\backslash\{(y_1, d_{y_1}), \dots, (y_j, d_{y_j})\}) \\
&\leq f(A^2),
\end{aligned}$$

where $j = 1$, $\{(y_1, d_{y_1}), \dots, (y_{j-1}, d_{y_{j-1}})\}$ is an empty set.

Theorem 1. *Algorithm 4 has an approximation ratio at least $\frac{1}{6}(1 - e^{-3}) \approx 0.15836$.*

Proof. Let T be the optimal solution with $|T| \geq 2$ and A be the seed set output by Algorithm 4. If $|A| < K$, then A is a $\frac{1-e^{-2}}{4}$-approximation solution of maximum k-submodular with a knapsack constraint [16]. Thus, A is also a $\frac{1-e^{-2}}{4}$-approximation solution of maximum our k-submodular objective function f with a knapsack constraint and a cardinality constraint.

If $|A| = K$, we distinguish between two cases based on the last iteration of the algorithm.

Case 1: For any node $x \in U(T)\backslash U(A)$, node $y \in U(A)\backslash U(T)$, the swap (x, y) was rejected because $\rho_{(x,y)} \leq 0$. Since A is a greedy solution, by Lemma 1, we derive

$$\begin{aligned}
f(T) &\leq 2f(A) + \sum_{x\in U(T)\backslash U(A)} [f(A \cup \{(x, d_x^*)\}) - f(A)] \\
&\leq 2f(A) + \sum_{x\in U(T)\backslash U(A)} [f(A \cup \{(x, d_x^*)\}\backslash\{(y(x), d_{y(x)})\}) - f(A\backslash\{(y(x), d_{y(x)})\})] \\
&\leq 2f(A) + \sum_{x\in U(T)\backslash U(A)} [f(A) - f(A\backslash\{(y(x), d_{y(x)})\})] \\
&\leq 3f(A),
\end{aligned}$$

where $(x, d_x^*) \in T$ and $y(x)$ is a one-to-one function with $(y(x), d_{y(x)}) \in A$. The second inequality holds because of submodularity. The third inequality holds as

$\rho_{(x,y)} \leq 0$. The last inequality holds by Lemma 2. Thus, in this case, we have Algorithm 4 yields a $\frac{1}{3}$-approximation on the optimum.

Case 2: At least one swap for node pair (x, y) with $\rho_{(x,y)} > 0$, for $x \in U(T)\backslash U(A)$, $y \in U(A)\backslash U(T)$ was rejected because $c(x) - c(y) + c(A) > B$.

Let A_t be the partial greedy solution after t iterations as shown in Algorithm 2 and 3. Let $l + 1$ be the first iteration in which a swap (a_{l+1}, b_{l+1}) was rejected since it violates the knapsack constraint where $a_{l+1} \in U(T)\backslash U(A_l)$. We can further assume that $l + 1$ is the first iteration t for which $A_t = A_{t-1}$. Since A_t is a greedy solution for $t = 0, 1, \ldots, l$, by Lemma 1, we have

$$f(T) \leq 2f(A_t) + \sum_{x \in U(T)\backslash U(A_t)} [f(A_t \cup \{(x, d_x^*)\}) - f(A_t)],$$

where $(x, d_x^*) \in T$.
For $t = 0, \ldots, K-1$, $|A_t| = t$, we have

$$f(T) \leq 2f(A_t) + \sum_{x \in U(T)\backslash U(A_t)} [f(A_t \cup \{(x, d_x^*)\}) - f(A_t)] \leq 2f(A_t) + B\rho_{t+1}.$$

The second inequality holds since $c(T) \leq B$.
For $t = K, \ldots, l$, $|A_t| = K$, we have

$$\begin{aligned} f(T) &\leq 2f(A_t) + \sum_{x \in U(T)\backslash U(A_t)} [f(A_t \cup \{(x, d_x^*)\}) - f(A_t)] \\ &\leq 2f(A_t) + \sum_{x \in U(T)\backslash U(A_t)} [f(A_t \cup \{(x, d_x^*)\}\backslash\{(y(x), d_{y(x)})\}) - f(A_t\backslash\{(y(x), d_{y(x)})\})] \\ &\leq 2f(A_t) + \sum_{x \in U(T)\backslash U(A_t)} [f(A_t \cup \{(x, d_x^*)\}\backslash\{(y(x), d_{y(x)})\}) - f(A_t)] + \\ &\quad \sum_{x \in U(T)\backslash U(A_t)} [f(A_t) - f(A_t\backslash\{(y(x), d_{y(x)})\})] \\ &\leq 3f(A_t) + \sum_{x \in U(T)\backslash U(A_t)} [f(A_t \cup \{(x, d_x^*)\}\backslash\{(y(x), d_{y(x)})\}) - f(A_t)] \\ &\leq 3f(A_t) + B\rho_{t+1}. \end{aligned}$$

The second inequality holds because of submodularity. The fourth inequality holds by Lemma 2. The last inequality holds since $c(T) \leq B$.
Therefore, for each $t = 0, 1, \ldots, l$, we have

$$f(T) \leq 3f(A_t) + B\rho_{t+1}.$$

Take advantage of the techniques in [17,18], we get

$$\frac{f(A_l\backslash\{(b_{l+1}, d_{b_{l+1}})\} \cup \{(a_{l+1}, d_{a_{l+1}})\})}{f(T)} \geq \frac{1}{3}(1 - e^{-3}),$$

where $\rho_{(a_{l+1},b_{l+1})} = \frac{f(A_l \backslash \{(b_{l+1},d_{b_{l+1}})\} \cup \{(a_{l+1},d_{a_{l+1}})\}) - f(A_l)}{c(a_{l+1})}$.
Therefore, **Singleton+Greedy-Local-Search** has a function value at least

$$\max\{f(A_l), f(\{(a_{l+1}, d_{a_{l+1}})\})\} \geq \frac{1}{2} f(A_l \cup \{(a_{l+1}, d_{a_{l+1}})\})$$
$$\geq \frac{1}{2} f(A_l \backslash \{(b_{l+1}, d_{b_{l+1}})\} \cup \{(a_{l+1}, d_{a_{l+1}})\}) \geq \frac{1}{6}(1 - e^{-3}) f(T).$$

The theorem is proved.

5 Conclusion

In this paper, we investigate the profit maximization problem for k kinds of competitive products information spreading at the same time in social networks. Considering that one seed user spreads multiple product information at the same time may disperse its followers' attention, one seed user can only propagate the influence of one kind of product. The goal of the proposed problem is to select k subsets of users as seeds with a budget B and a seed size K constraints such that the total profit for k kinds of products is maximized. Our optimal problem is formulated as maximizing a monotone k-submodular function under a knapsack constraint and a cardinality constraint. A Singleton+Greedy-Local-Search Algorithm is put forward in four steps to solve the profit maximization problem, which achieves a 0.216 and 0.158 approximation performance guarantees in two different cases, respectively.

Acknowledgment. This work is supported in part by the CCF-Huawei Populus Grove Fund under Grant No. CCF-HuaweiLK2022004, in part by the National Natural Science Foundation of China under Grant No. 62202109, No. 12101069 and in part by the Guangdong Basic and Applied Basic Research Foundation under Grant No. 2021A1515110321 and No. 2022A1515010611 and in part by Guangzhou Basic and Applied Basic Research Foundation under Grant No. 202201010676.

References

1. Ni, Q., Guo, J., Weili, W., Wang, H., Jigang, W.: Continuous influence-based community partition for social networks. IEEE Trans. Netw. Sci. Eng. **9**(3), 1187–1197 (2022)
2. Ni, Q., Guo, J., Weili, W., Wang, H.: Influence-based community partition with sandwich method for social networks. IEEE Trans. Comput. Soc. Syst. **10**(2), 819–830 (2023)
3. Kempe, D., Kleinberg, J., Tardos, É.: Maximizing the spread of influence through a social network. In: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 137–146. ACM (2003)
4. Bharathi, S., Kempe, D., Salek, M.: Competitive influence maximization in social networks. In: Deng, X., Graham, F.C. (eds.) WINE 2007. LNCS, vol. 4858, pp. 306–311. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-77105-0_31

5. Liang, Z., He, Q., Hongwei, D., Wen, X.: Targeted influence maximization in competitive social networks. Inf. Sci. **619**, 390–405 (2023)
6. Guanhao, W., Gao, X., Yan, G., Chen, G.: Parallel greedy algorithm to multiple influence maximization in social network. ACM Trans. Knowl. Discov. Data (TKDD) **15**(3), 1–21 (2021)
7. Zhang, H., Zhang, H., Kuhnle, A., Thai, M.T.: Profit maximization for multiple products in online social networks, pp. 1–9 (2016)
8. Chen, T., Liu, B., Liu, W., Fang, Q., Yuan, J., Weili, W.: A random algorithm for profit maximization in online social networks. Theor. Comput. Sci. **803**, 36–47 (2020)
9. Yuan, J., Tang, S.-J.: Adaptive discount allocation in social networks. In: Proceedings of the 18th ACM International Symposium on Mobile Ad Hoc Networking and Computing, pp. 1–10 (2017)
10. Huber, A., Kolmogorov, V.: Towards minimizing k-submodular functions. In: Mahjoub, A.R., Markakis, V., Milis, I., Paschos, V.T. (eds.) ISCO 2012. LNCS, vol. 7422, pp. 451–462. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32147-4_40
11. Ohsaka, N., Yoshida, Y.: Monotone k-submodular function maximization with size constraints. In: Advances in Neural Information Processing Systems, pp. 694–702 (2015)
12. Tang, Z., Wang, C., Chan, H.: On maximizing a monotone k-submodular function under a knapsack constraint. Oper. Res. Lett. A J. Oper. Res. Soc. Am. **50**(1), 28–31 (2022)
13. Wang, B., Zhou, H.: Multilinear extension of k-submodular functions. CoRR abs/2107.07103. https://arxiv.org/abs/2107.07103, p. eprint2107.07103 (2021)
14. Pham, C.V., Vu, Q.C., Ha, D.K.T., Nguyen, T.T., Le, N.D.: Maximizing k-submodular functions under budget constraint: applications and streaming algorithms. J. Comb. Optim. **44**, 723–751 (2022). https://doi.org/10.1007/s10878-022-00858-x
15. Ward, J., Živný, S.: Maximizing k-submodular functions and beyond. ACM Trans. Algorithms (TALG) **12**(4), 47 (2016)
16. Tang, Z., Chen, J., Wang, C.: An improved analysis of the greedy+singleton algorithm for k-submodular knapsack maximization. In: Li, M., Sun, X., Wu, X. (eds.) IJTCS-FAW 2023. LNCS, vol. 13933, pp. 15–28. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-39344-0_2
17. Sarpatwar, K.K., Schieber, B., Shachnai, H.: Constrained submodular maximization via greedy local search. Oper. Res. Lett. **47**(1), 1–6 (2019)
18. Sviridenko, M.: A note on maximizing a submodular set function subject to a knapsack constraint. Oper. Res. Lett. **32**(1), 41–43 (2004)