




Nonsubmodular Constrained Profit Maximization from Increment Perspective

Liman Du, Shengminjie Chen, Suixiang Gao, and Wenguo Yang^(✉) 

School of Mathematical Sciences,
University of Chinese Academy of Science, Beijing 100049, China
yangwg@ucas.edu.cn

Abstract. The growing importance of online social networks where people share information with others leads to the emergence of viral marketing, a new way to promote the sales of products. A derivation of classical Influence Maximization (IM) problem is the Profit Maximization (PM) problem that we focus on in this paper. We propose the PM problem with a cardinality constraint in order to make the problem closer to the real world. Without a fixed and pre-determined budget for seed selection, the profit spread metric of PM considers the total benefit and cost. The difference between influence spread metric and profit spread metric is that the latter is no longer monotone and lose the property of submodularity in general. Due to the natural form as the difference between two submodular functions, the profit spread metric admits a DS decomposition. What matters is that we design a Marginal increment-based Prune and Search (MPS) algorithm. From the perspective of marginal increment, MPS algorithm can compute profit spread more directly and accurately. Extensive experiments demonstrate the effective and outperformance of our algorithm.

Keywords: Profit maximization · Nonsubmodularity · Social network

1 Introduction

Due to their fast development, Online Social Networks (OSNs) become powerful mediums for spreading innovations. The market of OSNs advertisement is growing explosively. A managing partner of a consulting firm claimed in [1] that the e-commerce industry in the U.S. was worth approximately \$500 billion in 2018 and had been one of the fastest growing areas of the economy.

In OSNs, users can share their opinions about an event or a promoted product with other users, and this kind of information dissemination is in unprecedented prosperity nowadays. In the IM problem, a positive integer k is given and the problem aims to find a set of k seeds which maximizes the total number

Supported by National Natural Science Foundation of China under Grant No. 11991022 and No. 12071459. We would like to express our appreciation to the anonymous reviewers who have helped to improve the paper.

or expected number of active nodes in a social network. In [2], Kempe et al. firstly consider the issue of choosing influential sets of individuals as a discrete optimization problem, prove the problem is NP-hard under both Independent Cascade (IC) model and Linear Threshold (LT) model and propose a greedy algorithm which yields $(1-1/e)$ -approximation due to the submodularity and monotone properties of the influence spread. As shown in [3], computing exact influence in general networks under the LT model is $\#P$ -hard. Many follow-up, such as [4–9], further study IM problem in term of improving efficient implementation of algorithms, extending the maximization from a practical view and so on. When an OSN provider is hired to conduct the viral marketing campaign, it not only receives commission from the advertiser but also pay for the information propagation. Therefore, the OSN provider needs to account for both the benefit and cost of influence spread to maximize its profit. As an extension of IM, Tang J. et al. propose the PM problem whose aim is to find all the nodes that can maximize the profit in online social networks and define a general profit metric that can be expressed as the difference between benefit function and cost function in [11, 12]. Given that the benefit function is the total benefit brought by all the nodes activated, it is a submodular function. As for cost function, it is also submodular when it represents the total cost incurred by the whole active nodes, different from the budget constraint studied by previous works. Therefore, profit metric is a nonsubmodular and non-monotone function.

A lot of studies in [13–15] focus on submodular optimization while nonsubmodular optimization has been attracting more scholars' attention for many years. As summarized in [16], there are many approaches to solve nonsubmodular optimization problems. One of them is DS decomposition mentioned in [17–20]. As shown in [17, 18], every set function $f : 2^X \rightarrow R$ can be expressed as the difference of two monotone nondecreasing submodular functions g and h , i.e., $f = g - h$, where X is a finite set. Based on the theorem, many algorithms such as the modular-modular algorithm and iterated sandwich algorithm are proposed.

We observe some real marketing process and come to the conclusion that although the vast majority of companies select some users to help them promote products, the number of selected users is severely limited. The reason may be that excessive marketing not only bring profit but also incur greater cost. In this paper, we formulate the profit maximization problem with a cardinality constraint and accurately explain the profit function from the marginal increment perspective. The constrained profit maximization (CPM) problem aims to select at most k nodes such that the total profit generated by activated nodes is maximized. Different from existing common methods, we can obtain the marginal gain of profit from the perspective of marginal increment. Generally speaking, the profit spread function is neither submodular or monotone. Hence the CPM problem is a kind of nonsubmodular optimization problem. Due to its definition, profit function can be naturally expressed as the difference between two submodular functions. Therefore, we design a Marginal increment-based Prune and Search (MPS) algorithm. A algorithm whose goal is reducing the search

space is devised in the pruning phase and two algorithms inspired by the classic greedy algorithm and the DS decomposition method respectively are designed for selecting seed nodes.

The rest of this paper is organized as follows. In Sect. 2, we propose our formulation of the nonsubmodular CPM problem. Then, we derive MPS algorithm, a two-phase algorithm based on marginal increment method, in Sect. 3. Our experiment settings are introduced in Sect. 4 and results confirm the effectiveness of MPS algorithm. Concluding remarks and suggestions on the future works are given in Sect. 5.

2 Problem Formulation

2.1 Constrained Profit Maximization

In this section, we give CPM problem as follows. It should be emphasized that the marginal increment and CPM problem is not restricted by the diffusion model, and we take IC model for example. Given a directed graph $G = (V, E, P)$, a constant k , benefit b_v and cost c_v for each node $v \in V$, CPM problem aims to find a seed set S which includes at most k nodes to maximize the return profit.

We formulate the profit function from the perspective of marginal increment and describe the profit function value as accurate as possible. For the given directed graph $G = (V, E, P)$, nodes in V represent users and edges in E represent the connections among users. Each node v is associated with benefit b_v and cost c_v . For any directed edge $\langle u, v \rangle \in E$, refer to v as a neighbor of u and refer to u as an inverse neighbor of v . Denote $N_u = \{v : v \in V, \langle u, v \rangle \in E\}$. Let $p_{u,v}$, associated with each edge $\langle u, v \rangle \in E$, represents the activate probability from node u to node v . The diffusion process starts with a given set $S \subseteq V$ which includes all the active nodes. When a node u firstly becomes active, for each node $v \in N_u$, it has a single chance to activate v and succeeds with probability $p_{u,v}$. The diffusion process ends when there are no more nodes can be activated. Denoted $\phi(S)$ as the profit generated by a seed node set S . It is obvious that $\phi(S) = \beta(S) - \gamma(S)$, where $\beta(S)$ is the benefit of influence spread generated by a seed set S and $\gamma(S)$ is the total spread cost incurred by the whole nodes activated by S . The goal of constrained profit maximization problem is to find a seed set S satisfying $|S| \leq k$ to maximize the profit $\phi(S)$. Then, we will discuss more details about $\phi(S)$.

Property 1. $\phi(S) = \beta(S) - \gamma(S)$ is a non-submodular and non-monotone function in general.

2.2 Analysis of Profit Function

In this section, we will analyze the property of benefit function and cost function respectively from the perspective of marginal increment. The analysis will assist us reacquainting profit function.

Marginal Increment. Before discussing more details about benefit function and cost function, we recall some definitions about marginal increment provided in [21,22] as follows.

Definition 1. Suppose that $f : 2^V \rightarrow R$ is non-negative set function, where V is the ground set. For any subset A of V , $\Delta_v f(A) = f(A \cup \{v\}) - f(A)$ is called the marginal gain of $v \in V \setminus A$ at A . In addition, $\Delta_B f(A) = f(A \cup B) - f(A)$ is defined as the marginal gain of $B \subseteq V \setminus A$ at A .

It is well-known that for a non-negative set function $f : 2^V \rightarrow R^+$ defined on the ground set V , f is a monotone function if $f(A) \leq f(B)$ for any $A \subseteq B \subseteq V$ and is submodular, if $f(A) + f(B) \geq f(A \cup B) + f(A \cap B)$ for any $A, B \subseteq V$. According to above definitions, we can easily infer two properties. One is that for a given set $S \subseteq V$ and any subset $A \subseteq S$, $f(S) = f(S \setminus A) + \Delta_A f(S \setminus A) = f(A) + \Delta_{S \setminus A} f(A)$ for any given set function f . The other is that when f is monotone, $\Delta_A f(S \setminus A) \leq \Delta_B f(S \setminus B)$ and $\Delta_{S \setminus A} f(A) \geq \Delta_{S \setminus B} f(B)$ for any subset $A \subseteq B \subseteq S$.

Benefit Spread Function. At first, we formulate the benefit spread function from the perspective of marginal increment. According to the result of Jiang T. et al. in [11,12], it is obvious that benefit spread function $\beta(S)$ of influence spread generated by a seed set S is the benefit brought by all the nodes activated. Then, let $b_v^X \in (0, b_v]$ be the benefit spread of $v \in V$ with seed node set X , and for any $v \in V$, $b_v^\emptyset = 0$. Denote $\Delta b_v^X(u) = b_v^{X \cup \{u\}} - b_v^X$ as the marginal gain of the profit on node v when a new node u is selected as seed node. Then following formulas are proposed to calculate $\Delta b_v^X(u)$, the marginal gain of benefit spread function at node v .

Property 2. The marginal gain $\Delta b_v^X(u) = (b_v - b_v^X) p_{u,v} b_u^X$ for any $v \in N_u$ and for any $w \in N_v$, we have

$$\Delta b_w^X(u) = \frac{b_v - b_w^X}{1 - p_{(u,v)} b_v^X} p_{v,w} \Delta b_v^X(u). \tag{1}$$

Furthermore, if a node can be reachable from node u , we can update the marginal gain according to the topology order in recursive manner and we define $\Delta b_w^X(u) = 0$ for node w which is unreachable from node u .

Then, we can conclude that the objective function of benefit spread can be expressed as

$$\beta(X) = \sum_{v \in V} b_v^X = \sum_{v \in V} \sum_{u \in X} \Delta b_v^{X^u}(u), \tag{2}$$

where X^u is the set of nodes that have already activated before node u . Denote $X = \{v_1, v_2, \dots, v_{\hat{k}}\}$ as a node set which contains all the nodes that can be selected as seed, $\hat{k} = |X|$, $X^k = \{v_1, v_2, \dots, v_k\}$, $k = 1, 2, \dots, \hat{k}$ and $X^0 = \emptyset$ for convenience. Then $\beta(X)$ can be rewritten as $\beta(X) = \sum_{k=1}^{\hat{k}} \Delta_k \beta(X^{k-1})$, where $\Delta_k \beta(X^{k-1}) = \sum_{v \in V} \Delta b_v^{X^{k-1}}(v_k)$. We also have a property of $\beta(X)$ as follows.

Property 3. $\beta(X) = \sum_{v \in V} \sum_{u \in X} \Delta b_v^{X^u}(u) = \sum_{k=1}^{\widehat{k}} \Delta_k \beta(X^{k-1})$ is submodular and monotone decrease with $b_v^{X^{k-1}}$, for any $v \in V$ and $k = 1, 2, \dots, \widehat{k}$

Cost Function. We have formulated and analyzed the property of benefit spread function, and it is time to turn to the cost spread function. Generally speaking, the cost function changes during the spread process. The cost of influence propagation induced by a seed set X is the total cost incurred by all the nodes activated. Therefore, from the marginal increment perspective, let c_v^X be the cost spread of $v \in V$ with seed node set X . Denote $\Delta c_v^X(u) = c_v^{X \cup \{u\}} - c_v^X$ as the marginal gain of the cost on node v when a new node u is selected as seed node, and for any $v \in V$, if $v \notin X$, $c_v^\emptyset = 0$. We formulate the cost spread function as

$$\gamma(X) = \sum_{v \in V} c_v^X = \sum_{v \in V} \sum_{u \in X} \Delta c_v^{X^u}(u), \tag{3}$$

where X^u is the set of nodes that have already activated before node u . Let $X = \{v_1, v_2, \dots, v_{\widehat{k}}\}$ represent the node set that includes all the candidate nodes, $\widehat{k} = |X|$, $X^k = \{v_1, v_2, \dots, v_k\}$, $k = 1, 2, \dots, \widehat{k}$ and $X^0 = \emptyset$. Then $\gamma(X) = \sum_{k=1}^{\widehat{k}} \Delta_k \gamma(X^{k-1})$, where $\Delta_k \gamma(X^{k-1}) = \sum_{v \in V} \Delta c_v^{X^{k-1}}(v_k)$. We can see that another property of cost function $\gamma(X)$ can be proposed as follows.

Property 4. $\gamma(X) = \sum_{k=1}^{\widehat{k}} \Delta_k \gamma(X^{k-1})$ is submodular and monotone decrease with $c_v^{X^{k-1}}$, for $v \in V$, and $k = 1, 2, \dots, \widehat{k}$.

3 Algorithm for Constrained Profit Maximization Problem

We devise a marginal increment-based two-phase algorithm MPS for CPM problem. One of the highlights of the MPS algorithm is that benefiting from the method of marginal increment, we can compute the benefit spread function and cost spread function more accurately. At the first phase of MPS, we use Modified Iterative Prune (MIP) algorithm to reduce search space. Then, we can select seed nodes by one of the two different algorithms proposed in Sect. 3.3.

3.1 Marginal Increment Computation

First of all, we consider $\phi(X) = \beta(X) - \gamma(X)$, where $\beta(X)$ and $\gamma(X)$ is the total benefit and cost of all the nodes activated respectively. From the perspective of marginal increment, for any $X \subseteq V$ only containing seed nodes, the benefit and cost can be expressed as $\beta(X) = \sum_{v \in V} b_v^X = \sum_{v \in V} \sum_{u \in X} \Delta b_v^{X^u}(u)$, $\gamma(X) = \sum_{v \in V} c_v^X = \sum_{v \in V} \sum_{u \in X} \Delta c_v^{X^u}(u)$. We design two algorithms to compute $\beta(X)$ and $\Delta \beta^X(u)$ respectively. And the method for computing $\gamma(X)$ and $\Delta \gamma^X(u)$ is the same as that for computing $\beta(X)$ and $\Delta \beta^X(u)$. Inspired by some existing studies of nonsubmodular optimization, there are several ways to deal

with $\phi(X)$. It should be emphasized that this profit spread metric can be viewed as the difference between two submodular functions. Therefore, algorithm based on DS decomposition can be used naturally. We propose a two-phase framework to solve the CPM problem, which includes pruning phase and search phase.

Algorithm 1. Marginal Increment for computing $\beta(X)$

Input: $G(V, E)$, IC model and candidate set $X = \{u_1, u_2, \dots, u_k\}$, $b_v^{X^i}$ for any $v \in V$
Output: function value $\beta(X)$
 Initialize $i = 0$ and $b^{X^i} = \{\dots, b_v^{X^i}, \dots\}$
for $u_i \in X$ **do**
 update $b^{X^i} = \{\dots, b_v^{X^i}, \dots\}$ with $b_v^{X^i} = b_v^{X^{i-1}} + (1 - b_v^{X^{i-1}}) p_{u_i, v} b_{u_i}^{X^{i-1}}$ for each
 $v \in V$ according to the topological order
end for
 $\beta(X) = \sum_{v \in V} b_v^{X^k}$
return $\beta(X)$

Algorithm 2. Marginal Increment for computing $\Delta\beta^X(u)$

Input: a reference set $X \subseteq V$ and a node $u \in V \setminus X$
Output: marginal gain $\Delta\beta^X(u)$
for $v \in V$ **do**
 compute $\Delta b^X(u) = \{\dots, \Delta b_v^X(u), \dots\}$ with $\Delta b_v^X(u) = (1 - b_v^X) p_{u, v} b_u^X$ according
 to the topological order
end for
 $\Delta\beta^X(u) = \sum_{v \in V} \Delta b_v^X(u)$
return $\Delta\beta^X(u)$

3.2 Pruning Phase

Considering the constraints of PM problem mentioned in [12], the marginal profit gain is bounded below by the smallest benefit gain less the largest cost gain and bounded above by the largest benefit gain less the smallest cost gain. Apparently, for any node $v \in V$, if the lower bound of marginal profit gain is positive, v may be selected in an optimal solution of PM. Similarly, a node v cannot be selected in an optimal solution when its marginal profit gain is bounded up by a negative number. Starting with $A_0 = \emptyset$, $B_0 = V$, $t = 0$, algorithm mentioned in [12] extends the idea in an iterative manner and return A_t as A^* , B_t as B^* when they are converged. It is proved that for any global maximizer S^* of a unconstrained profit problem, it holds that $A_t \subseteq A_{t+1} \subseteq A^* \subseteq S^* \subseteq B^* \subseteq B_{t+1} \subseteq B_t$ for any $t \leq 0$. Therefore, only the nodes in $B^* \setminus A^*$ need to be further examined for seed

Algorithm 3. Modified Iterative Prune (MIP)

```

start with  $A_0 \leftarrow \emptyset$ ,  $B_0 \leftarrow B$  and  $t = 0$ 
repeat
   $A_{t+1} \leftarrow A_t \cup \{u : \Delta\beta^{B_t \setminus \{u\}}(u) - \Delta\gamma^{A_t}(u) > 0 \text{ and } u \in B_t \setminus A_t\}$ 
   $B_{t+1} \leftarrow B_t \setminus \{u : \Delta\beta^{A_t}(u) - \Delta\gamma^{B_t \setminus \{u\}}(u) < 0 \text{ and } u \in B_t \setminus A_t\}$ 
   $t \leftarrow t + 1$ 
until  $A_t = A_{t+1}$  and  $B_t = B_{t+1}$ 
return  $A_t$  as  $A^*$ ,  $B_t$  as  $B^*$ 

```

selection. However, when a cardinality constraint is added, the problem becomes more difficult. In order to make the computation easier, we remove some nodes in V to obtain a node set $B \subset V$ satisfying the condition that the size of $S^* \setminus B$ is as small as possible. And based on it, we design Modified Iterative Prune (MIP) algorithm. It requires skill to choose an appropriate start baseline B . Many methods mentioned in [23] can be studied further more. In the experiments, we adopt two different strategies and compare their performance.

3.3 Search Phase

Considering the natural DS decomposition form of profit function, we propose two algorithms in the search phase. One practical algorithm is the Marginal Increment Greedy Algorithm shown as Algorithm 4. The other available method is the Improved Modular-Modular algorithm given as Algorithm 5.

Marginal Increment Greedy Algorithm. Given a directed graph $G(V, E)$ where V represents the whole user and E is the set of their relations. After conducting MIP algorithm, we can obtain A^* and B^* . As above discussion reveals, all the nodes in A^* has a non-negative marginal profit gain and may be selected as seed node for CPM problem while every $v \in V \setminus B$ cannot be contained in any optimal seed node set. If $|A^*| \leq k$, the seed node set X is initialized as A , else $X = \emptyset$. In each iteration, MIGA algorithm selects node u which has the largest marginal profit gain $\Delta\phi^X(u) = \Delta\beta^X(u) - \Delta\gamma^X(u)$ and adds it into X . Repeat the process until no node left has positive marginal gain or $|X| > k$. In the meanwhile, we propose another algorithm, which is based on the ModMod procedure and also be combined with MIP algorithm to solve the CPM problem.

Improved Modular-Modular Algorithm. The ModMod procedure introduced by Iyer and Bilmes in [17] aims to optimize set function expressed as the difference between submodular functions. And given that only nodes in B^* should be considered for CPM, we can obtain two modular upper bounds of $\gamma(Y)$ which are both tight at a given set X^t for any $Y \subseteq V$ as

Algorithm 4. Marginal Increment Greedy Algorithm (MIGA)

Input: $G(V, E)$, a constant k, A^*, B^*
Output: a set of k nodes $X \subseteq B^*$

```

if  $|A^*| \leq k$  then
     $X = A^*$ 
else
     $X = \emptyset$ 
end if
while  $|X| \leq k$  do
     $u \leftarrow \arg \max_{u \in B^*} (\Delta\beta^X(u) - \Delta\gamma^X(u))$ 
    update  $\Delta\beta^X(u)$  and  $\Delta\gamma^X(u)$ 
    if  $\Delta\beta^X(u) - \Delta\gamma^X(u) \leq 0$  then
        return  $X$ 
    end if
     $X \leftarrow X \cup \{u\}$ 
end while
return  $X$ 

```

$$\gamma(Y) \leq m_{X^t}^1(Y) = \gamma(X^t) - \sum_{u \in X^t \setminus Y} \Delta\gamma^{B^* \setminus \{u\}}(u) + \sum_{u \in Y \setminus X^t} \Delta\gamma^{X^t}(j), \quad (4)$$

$$\gamma(Y) \leq m_{X^t}^2(Y) = \gamma(X^t) - \sum_{u \in X^t \setminus Y} \Delta\gamma^{X^t \setminus \{u\}}(u) + \sum_{u \in Y \setminus X^t} \Delta\gamma^\emptyset(j). \quad (5)$$

For convenience, we denote $m_{X^t}(Y)$ represent above two tight upper bound, i.e., $m_{X^t}(Y)$ can be explained as $m_{X^t}^1(Y)$ or $m_{X^t}^2(Y)$. Let π be any permutation of V that places all the nodes in $X^t \subseteq V$ before the nodes in $V \setminus X^t$. Let $S_i^\pi = \{\pi(1), \pi(2), \dots, \pi(i)\}$ be a chain formed by the permutation, where $S_0^\pi = \emptyset$ and $S_{|X^t|}^\pi = X^t$. Define $h_{X^t}^\pi(\pi(i)) = \beta(S_i^\pi) - \beta(S_{i-1}^\pi)$. Then, $h_{X^t}^\pi(Y) = \sum_{v \in Y} h_{X^t}^\pi(v)$ is a lower bound of $\beta(Y)$ for any $Y \subseteq V$ tight at X^t .

Algorithm 5. Improved Modular-Modular Algorithm (IMM)

```

Initialize  $t \leftarrow 0$ 
if  $|A^*| \leq k$  then
     $X^0 = A^*$ 
else
     $X^0 = \emptyset$ 
end if
repeat
    choose the permutations of  $X^0, X^t \setminus X^0, B^* \setminus X^t$  and concatenate them as  $\pi$ 
     $X^{t+1} \leftarrow \arg \max_{Y \subseteq B^*} h_{X^t}^\pi(Y) - m_{X^t}(Y)$ 
     $t \leftarrow t + 1$ 
until  $|X^t| > k$  or  $X^t = X^{t-1}$ 
Return  $X^t$ 

```

4 Experiments

In this section, we conduct experiments on three data sets to test the effectiveness of MPS algorithm for optimizing ϕ , and compare it with other different algorithms. When cost function represents the total cost of seed nodes, i.e. $\phi(X) = \sum_{v \in V} b_v^X - \sum_{u \in X} c(u)$, we can view it as a special case of $\phi(X)$.

4.1 Experiment Setup

A synthetic graph and two real-world social graphs are used in our experiment. We will describe them more precisely in the following. We can see a wide variety of relationship can be represented by these social graphs.

Synthetic: This is a relatively small acyclic directed graph randomly generated with 2708 nodes and 5278 edges.

Facebook: The Facebook data set consists of 4039 users and 88234 edges collected from survey participants and has been anonymized.

Wikipedia: The Wikipedia data set is generated by a voting activity, containing 7115 nodes and 103689 edges. Nodes in the network represent wikipedia users and a directed edge from node i to node j represents that user i voted on user j .

All the data sets come from Stanford Large Network Dataset Collection and only have the relationship between two nodes. We use the Independent Cascade propagation model. For ease of comparison, we have some assumption as follows. For the propagation probability, we use a trivalency model in [21], selecting a value from (0.1, 0.01, 0.001) at random. And the profit of each node is set to be 1. The strategies we use in the experiment include:

Random: It randomly selects k nodes. We run the algorithm 10 times and take their average value as the expected profit.

MaxDegree: We select top k nodes according to their degree.

MGIA: We conduct the MIGA algorithm where $A^* = \emptyset$ and $B^* = V$.

MPS with B^1 : It includes two phases. MIP algorithm is conducted and the MIGA is carried out subsequently. In the pruning phase, all the nodes are sorted according to their degree and top 500 nodes is included in B .

MPS with B^2 : This algorithm is similar to MPS with B^1 mentioned above while each node in the start baseline B has at least two adjacent edges.

The algorithms are all implemented in MATLAB and the experiments are carried out on a machine with an Inter Core i5-6200U 2.30 GHz CPU and 8 GB memory. The running time of MPS algorithm is greatly sensitive to the choice of B and under above setup the MPS algorithm takes several hours in average.

4.2 Exprement Result

In this section, we value the effectiveness of MPS. First of all, we compare the effect of two different start baseline. Then, we show the relation of cost and profit spread value. The last but not least is the comparison with other seed nodes selection algorithms.

In Fig. 1, we can see that different start baselines can produce different effect on the effectiveness of MPS algorithm. Due to the different definitions of B^1 and B^2 , it is inevitable that more potential seed nodes are excluded from B^1 , resulting in a small profit spread value in Facebook and Wikipedia. And in Fig. 2, it is obvious that the change speed of profit spread value decreases with the increasing cost. Judging from this result, the constraint of seed node set's size is important when the cost is not small. The Fig. 3 illustrates that MPS outperforms other algorithms excluding MIGA. And the difference between MIGA and MPS with B^2 is small. It should be noted that the difference is smaller in Wikipedia network than in Facebook network. Therefore, MPS may perform better in a network which contains more nodes and edges. We can also arrive at a conclusion that MPS will perform better with the a more ideal start baseline B_0 .

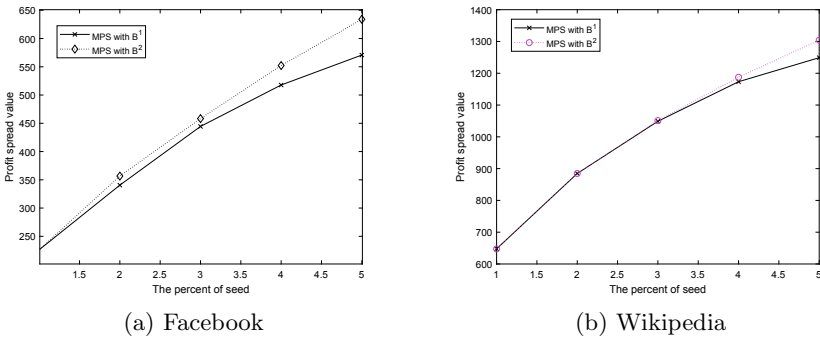


Fig. 1. Profit spread value versus different prune start baseline

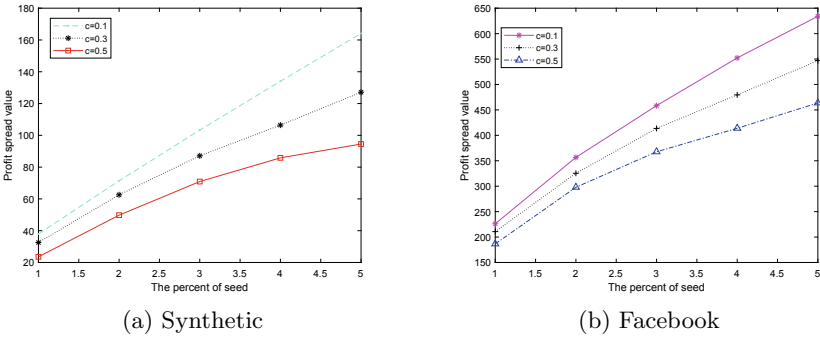


Fig. 2. Profit spread value versus different cost

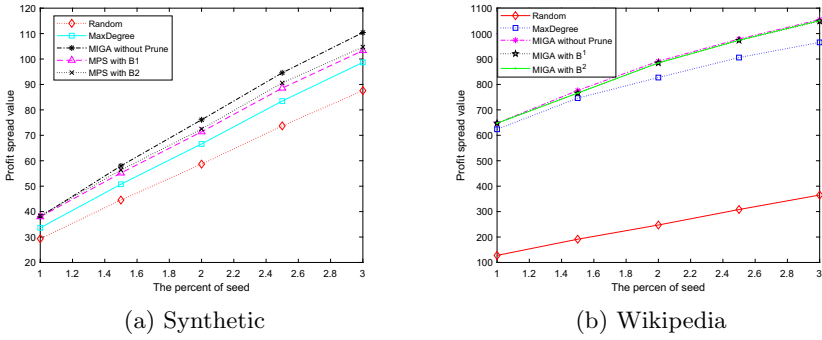


Fig. 3. Comparison with other methods

5 Conclusion and Future Works

In this paper, we have studied the CPM problem and formulate it from an incremental marginal gain perspective. Given that the objective function of CPM problem lacks submodularity in general, we design MPS algorithm to optimize the profit function. In the first phase, MIP algorithm is used to reduce the search space. The MIGA and IMM algorithm are devised in the second phase to select at most k seed nodes. Based on the marginal increment method, these algorithms calculate the profit function as accurate as possible. Experimental results show that our MPS algorithm substantially outperform some other algorithms, and it is also perform well with submodular profit metric after a few adjustments.

Several research directions of CPM problem deserve further study. The first and most problem is how to select the baseline of MIP algorithm. We can see that B in MIP algorithm directly affects the algorithm’s efficiency and performance and MPS algorithm is feasible in large-scale social networks with a ideal B . Therefore, the property of a ideal baseline B need to be studied. Beyond that, more efforts on study whether MPS algorithm can generate a constant approximation is significant.

References

1. George D.: Google, Facebook and Amazon are the only winners in Ecommerce. <https://www.entrepreneur.com/article/328478>. Last accessed 5 Sep 2020
2. Kempe, D., Kleinberg, J., Tardos, E.: Maximizing the spread of influence through a social network. In: 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 137–146. ACM, New York (2003)
3. Chen, W., Wang, C., Wang, Y.: Scalable influence maximization for prevalent viral marketing in large-scale social networks. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1029–1038. ACM, New York (2010)

4. Kempe, D., Kleinberg, J., Tardos, É.: Influential nodes in a diffusion model for social networks. In: Caires, L., Italiano, G.F., Monteiro, L., Palamidessi, C., Yung, M. (eds.) ICALP 2005. LNCS, vol. 3580, pp. 1127–1138. Springer, Heidelberg (2005). https://doi.org/10.1007/11523468_91
5. Kimura, M., Saito, K.: Tractable models for information diffusion in social networks. In: Fürnkranz, J., Scheffer, T., Spiliopoulou, M. (eds.) PKDD 2006. LNCS (LNAI), vol. 4213, pp. 259–271. Springer, Heidelberg (2006). https://doi.org/10.1007/11871637_27
6. Even-Dar, E., Shapira, A.: A note on maximizing the spread of influence in social networks. In: Deng, X., Graham, F.C. (eds.) WINE 2007. LNCS, vol. 4858, pp. 281–286. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-77105-0_27
7. Borgs, C., Brautbar, M., Chayes, J., et al.: Maximizing social influence in nearly optimal time. In: 25th ACM-SIAM Symposium on Discrete Algorithms, pp. 946–957. Society for Industrial and Applied Mathematics, USA (2014)
8. Youze, T., Xiaokui, X., Yanchen, S.: Influence maximization: near-optimal time complexity meets practical efficiency. In: the 2014 ACM SIGMOD International Conference on Management of Data (SIGMOD 2014), pp. 75–86. ACM, New York (2014)
9. Youze, T., Yanchen, S., Xiaokui, X.: Influence maximization in near-linear time: a martingale approach. In: The 2015 ACM SIGMOD International Conference on Management of Data (SIGMOD 2015), pp. 1539–1554. ACM, New York (2015)
10. Goyal, A., Lu, W., Lakshmanan, L.V.S.: SIMPATH: an efficient algorithm for influence maximization under the linear threshold model. In: 13th International Conference on Data Mining, Vancouver, BC, pp. 211–220. IEEE (2011)
11. Tang, J., Tang, X., Yuan, J.: Profit maximization for viral marketing in online social networks: algorithms and analysis. *IEEE Trans. Knowl. Data Eng.* **30**(6), 1095–1108 (2018)
12. Tang, J., Tang, X., Yuan, J.: Towards profit maximization for online social network providers. [arXiv:1712.08963](https://arxiv.org/abs/1712.08963) (2017)
13. Lee, J., Mirrokni, V., Nagarajan, V., Sviridenko, M.: Maximizing nonmonotone sub-modular functions under matroid or knapsack constraints. *SIAM J. Disc. Math.* **23**(4), 2053–2078 (2010)
14. Gupta, A., Roth, A., Schoenebeck, G., Talwar, K.: Constrained non-monotone submodular maximization: offline and secretary algorithms. In: Saberi, A. (ed.) WINE 2010. LNCS, vol. 6484, pp. 246–257. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-17572-5_20
15. Feldman, M., Naor, J., Schwartz, R.: A unified continuous greedy algorithm for submodular maximization. In: 52nd Annual Symposium on Foundations of Computer Science (FOCS 2011), USA, pp. 570–579. IEEE Computer Society (2011)
16. Wu, W., Zhang, Z., Du, D.Z.: Set function optimization. *J. Oper. Res. Soc. China* **7**(2), 183–193 (2019)
17. Iyer, R., Bilmes, J.: Algorithms for approximate minimization of the difference between submodular functions, with applications. In: 28th Conference on Uncertainty in Artificial Intelligence (UAI 2012), Arlington, Virginia, pp. 407–417. AUAI Press (2012)
18. Narasimhan M., Bilmes J.: A submodular-supermodular procedure with applications to discriminative structure learning. In: 21st Conference on Uncertainty in Artificial Intelligence (UAI 2005). [arXiv:1207.1404](https://arxiv.org/abs/1207.1404)

19. Bai W., Bilmes J.: Greed is still good: maximizing monotone submodular + supermodular functions. In: Dy, J., Krause, A. (eds.) 35th International Conference on Machine Learning (ICML 2018). LNCS, vol. 80. pp. 304–313. PMLR (2018)
20. Li, X., Du, H.G., Pardalos, P.M.: A variation of DS decomposition in set function optimization. *J. Comb. Opt.* **40**(1), 36–44 (2020). <https://doi.org/10.1007/s10878-020-00560-w>
21. Yang, W., et al.: Marginal gains to maximize content spread in social networks. *IEEE Trans. Comput. Soc. Syst.* **6**(3), 479–490 (2019)
22. Yang, W., Zhang, Y., Du, D.-Z.: Influence maximization problem: properties and algorithms. *J. Comb. Opt.* **40**(4), 907–928 (2020). <https://doi.org/10.1007/s10878-020-00638-5>
23. Erkol, Ş., Castellano, C., Radicchi, F.: Systematic comparison between methods for the detection of influential spreaders in complex networks. *Sci. Rep.* **9**, 15095 (2019). <https://doi.org/10.1038/s41598-019-51209-6>