Profit Maximization for Multiple Products in Online Social Networks

Huiyuan Zhang, Huiling Zhang, Alan Kuhnle, and My T. Thai
Department of Computer and Information Science and Engineering
University of Florida, Gainesville, Florida 32611
Email: {huiyuan, huiling, kuhnle, mythai}@cise.ufl.edu

***Abstract*—Information propagation in online social networks (OSNs), which helps shaping consumers' purchasing decisions, has received a lot of attention. The ultimate goal of marketing and advertising in OSNs is to massively influence audiences and enlarge the number of product adoptions. Most of existing works focus on maximizing the influence of a single product or promoting the adoption of one product in competing campaigns. However, in reality, the majority of companies produce various products for supplying customers with different needs. Therefore, it is truly significant and also challenging to wisely distribute limited budget across multiple products in viral marketing.**

In this paper, we investigate a Profit Maximization with Multiple Adoptions (PM^{2}A) problem, which aims at maximizing the overall profit across all products. The natural greedy fails to provide a bounded result. In order to select high quality seeds for information propagation, we first proposed the PMCE algorithm, which has a ratio $\frac{1}{2}(1-1/e^2)$. Moreover, we further improve this ratio to $(1-1/e)$ by proposing the PMIS algorithm. Comprehensive experiments on three real social networks are conducted. And results show that our algorithms outperform other heuristics, and better distribute the budget in terms of profit maximization.

***Index Terms*—Viral Marketing, Social Networks, Approximation Algorithm**

I. Introduction

In recent decades, influence spreads rapidly through social interactions, especially when facilitated by modern technology, including blogs and online social networking sites. Recent statistical results show that OSNs like YouTube, Facebook and LinkedIn are among top-ten most visited websites on the Internet [1], and thus providing a nice platform for fast information propagation and frequent human interactions. Such interactions including product advertisements, recommendations can directly influence people's behaviors. Moreover, recent studies support that network-based marketing has a direct influence on increasing product adoption. Notable examples include Rokenbok, a toy company gets 50% of its customers generated from YouTube [2], and State Bicycle, a bicycle company drives close to $500,000 in incremental sales annually through Facebook advertisement [3].

Motivated by the huge impact of word-of-mouth effect on economy, massive studies about viral marketing have been devoted to the question how to select the most influential seeding set in order to widely spread influence [4]–[9]. Most of recent advanced research focuses on two essential aspects. One is aiming at modeling the underlying propagation process more precisely and practically, where various extensions [5], [10]–[12] are made from the two basic diffusion models, namely *Linear Threshold* model and *Independent Cascading* model [4]. The other is targeting on improving efficiency and scalability of algorithms [6], [13], given that nowadays OSNs are growing at a phenomenal rate.

Most of above mentioned works study influence maximization problem for a single diffusion, which indicates that there is only one considered product. However, in reality, most companies do not just produce one item. Instead, they usually maintain several different production lines to supply customers with various demands. For example, Samsung produces both cheap ordinary phones and expensive smart phones; Intel produces normal CPU for personal computers and high end CPU for intensively used servers. Moreover, consumer buying power keeps growing day by day. For products with fast consumption and large demands like food and clothes, people probably purchase multiple quantities at a time. For some products that are expensive and have long lifetime of usage, people still like to adopt more due to their different functionaries. Examples include that many people own iPhone and iPad at the same time, and that many people have both laptop and desktop. Therefore, from the perspective of companies, profit maximization of multiple products is of great importance. Given a limited budget and several items with different producing costs and profits, it is crucial to wisely allocate the budget for the sake of maximizing overall profit.

In order to design the optimal marketing strategy, we formulate a *Profit Maximization of Multiple Adoption* (PM^{2}A) problem, which seeks for a seeding set within limited budget to massively influence users and achieve the goal of profit maximization. The challenges of solving PM^{2}A come from following aspects. First, a person may have diverse propensities for products with different attributes and can adopt multiple products. Second, different people exhibit heterogeneous adoption behaviors for the same product. Most importantly, the given budget is shared by all products which results in the essential relation among marketing strategies for each product. Thus, existing solutions to the profit maximization of a single product cannot be directly applied to cases where there are multiple products, which motivates us to devote to the design of effective solutions to PM^{2}A. For the special case, where all products are of same marketing cost, we can apply a

Huiyuan Zhang and Huiling Zhang contributed equally to this work.

978-1-4673-9953-1/16/$31.00 ©2016 IEEE

naive greedy but it does not produce bounded solutions when seeding costs for products are not uniform. For general cases, we propose other approximation algorithms.

Our contributions in this paper are summarized as follows:

- This is the first attempt to study the profit maximization with multiple products and adoptions in OSNs. More than one diffusion propagates upon the network. Both uniform seeding costs and different costs are considered.
- We propose a PMCE algorithm, which maintains a ratio of $\frac{1}{2}(1-1/e^2)$ to the optimal solution. Moreover, by designing PMIS algorithm, we further improve the ratio to $(1-1/e)$. Meanwhile, any existing technique on improving running time of influence maximization problem can be applied here, and this algorithm enables parallel processing for all products, which further improves its scalability in large-scale networks.
- Experimental study is conducted on three real-world datasets, and results show that both PMCE and PMIS, which are approximation algorithms, outperform other heuristics, and the total profit achieved is significantly improved.

Organization. The rest of this paper is organized as follows. Section II introduces the information diffusion model and gives the problem definition. In Section III, we elaborate two proposed approximation algorithms along with detailed analysis about their ratios and time complexity. The experimental evaluation results are demonstrated in Section IV. Section V discusses future research work and Section VI reviews the related work. Finally, this paper is concluded in Section VII.

II. Diffusion Models and Problem Definition

A. Multiple Thresholds Model

The focus of this paper is to design marketing strategy for budget allocation. In order to tackle this problem, we need a diffusion model as an input to investigate into the propagation process on the underlying network. There are many diffusion models in the literature. The most well known models are the Linear Threshold (LT) model and the Independent Cascading (IC) model [4]. However, they are not suitable for multiple products. Considering various attributes of products as well as user behavior in making adoptions, we extend from LT model and present a Multiple Thresholds (MT) model as follows.

The social network is represented as a directed graph $G = (V, E)$ where nodes in V represent users and edges represent social interactions between users. Each node in G can choose multiple status from $q+1$ states $\{0, 1, 2, ..., q\}$ indicating its adoption behavior, where q is the number of product categories. A node u is considered to be active if at least one of its states $i > 0$, which indicates the corresponding product has been adopted. A node is at state 0 when it does not make any adoptions, and we can also say it is inactive in this case. Each node $v \in V$ is associated with a threshold vector $\Theta_v = (\theta_v^1, \theta_v^2, ..., \theta_v^q)$, where $\theta_v^i \in [0,1], 1 \le i \le q$ is the threshold of node v corresponding to product i. Every edge $(u,v) \in E$ is assigned a real value weight $w_{uv} \in [0,1]$ such that $\sum_v w_{uv} \le 1$, which represents the influence from u on v. Furthermore, we denote u's incoming neighbors (in-neighbors) and outgoing neighbors (out-neighbors) as $N^+(u)$ and $N^-(u)$, respectively.

The diffusion process on the MT model is as follows. Given threshold vectors and an initial set of active nodes S, the diffusion process unfolds deterministically in discrete steps: in time step t, all nodes that adopted product i in step $t-1$ stay in the same state, and activate any node v for which the total weight of its active neighbors exceeds its corresponding threshold θ_v^i, i.e, $\sum_{u \in \sigma^i \cap N^+(v)} w_{uv} \ge \theta_v^i$. The diffusion terminates when there is no more new activation occurs. Note that node v can adopt multiple products if there are more than one threshold has been satisfied.

B. Problem Definition

For a seed set S, let $\sigma^i(S)$ denote the set of users who adopt product i and $\rho^i(S)$ denote the total profit achieved by adoptions of product i. Moreover, $\sigma(S)$ and $\rho(S)$ denote the number of overall activated users and total profit gained by adoptions of all kinds of products, respectively.

Definition 1 (Profit Maximization with Multiple Adoptions (PM^{2}A))**.** *Given a directed graph $G = (V, E)$ with vertices in V representing the individuals in the network and edges in E modeling the social relationship between individuals, and a positive real number B. Assume there are q different kinds of products in total, and let c_i be the cost to initially activate a node to adopt product i, and p_i denote the profit that can be gained when a node adopts product i. The PM^{2}A problem asks to identify a set of nodes S with overall activation cost at most B such that the total achieved profit $\rho(S)$ is maximized.*

Note that, when there is only one product, the PM^{2}A problem is equivalent to the traditional influence maximization problem under the LT model, which has been shown to be NP-hard [4]. Therefore, PM^{2}A problem is also NP-hard.

III. Solution

Intuitively, we can have a native greedy solution, which iteratively chooses a node with the maximum marginal gain into the seeding set with the consideration of costs and profits. That is, pick a node v maximizing the ratio of $\frac{\delta_v(S)}{c(v)}$, where $\delta_v(S) = \rho(S \cup \{v\}) - \rho(S)$. In the case where all products have uniform cost, this heuristic has a performance guarantee.

Theorem 1. *The native greedy algorithm achieves a ratio of $(1-1/e)$ for the PM^{2}A problem with uniform activation costs.*

Proof. For each node $v \in V$, there is a threshold vector Θ_v associates with it. Instead of studying the diffusion process in a single network G with multiple thresholds for nodes, we can view the spreading process as unfolding on q separate networks $G^i = (V, E, \theta^i)$ with threshold corresponding to each product. Since node $v \in V$ will be activated by product i whenever $\sum_{u \in N^- \cap \sigma^i(S)} w_{uv} \ge \theta_v^i$, and is able to adopt more than one product when there are several thresholds have been satisfied. Thus, the total profit is the summation of profit over all networks $\rho(S) = \sum_{i=1}^q \rho^i(S)$. Since $\rho^i(S) = p_i \sigma^i(S)$, and

$\sigma^i(S)$ has been shown to be monotone increasing as well as submodular [4], then the total profit $\rho(S) = \sum_{i=1}^{q} p^i \sigma^i(S)$, which is the linear combination of $\sigma^i(S)$, is also monotone increasing and submodular. Hence, the greedy algorithm is guaranteed to find a solution which achieves at least a constant fraction $(1 - 1/e)$ of the optimal profit. □

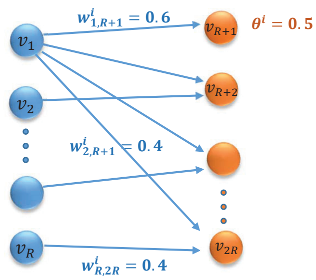


Fig. 1: A naive greedy cannot return bounded solutions.

However, with different seeding costs, the native greedy can perform arbitrarily bad. For instance, given a budget B, we want to promote three products $\{1, 2, 3\}$, with $c_1 = \epsilon_1$, $p_1 = \epsilon_1$, $c_2 = B - \epsilon_2$, $p_2 = B - \epsilon_1$, $c_3 = \epsilon_2$, $p_3 = \epsilon_1$, where ϵ_1 and ϵ_2 are small positive numbers with $\epsilon_1 > \epsilon_2$. Fig. 1 shows a bipartite graph $G = (V, E)$, where vectors on edges indicate influence and vectors associated with nodes are adoption thresholds with respect products.

Obviously, by activating node v_1, we can have gain $\rho^1(\{v_1\}) = \epsilon_1 R$, $\rho^2(\{v_1\}) = (B - \epsilon_1)R$ and $\rho^3(\{v_1\}) = \epsilon_1 R$. Applying the naive greedy, v_1 is first selected as a seed of product $\{3\}$, followed by being chosen for product $\{1\}$, and then, due to limited budget, we cannot make more activation. Therefore, profit $(\epsilon_1 + \epsilon_2)R$ is obtained. However, the optimal solution is letting node u be the initiator of products $\{2, 3\}$, which ends up with profit BR. It can be see that natural greedy fails to return bounded solutions for the PM^{2}A problem. In the following part, we propose two algorithms to handle this problem and effectively return theoretical bounded solutions.

A. *Profit Maximization with Cost Effectiveness*

Though the aforementioned naive greedy differentiates costs, it fails to provide bounded solutions when the seeding costs are not uniform. In this subsection, we present a Profit Maximization with Cost Effectiveness (PMCE) algorithm, which can provide a theoretical performance bound for general activation costs. Specifically, in PMCE, we obtain two candidate sets by following two different directions to modify the greedy rule. Then we select the better one from these two sets as the final solution.

The first direction of modification is to emphasize the importance of product cost. We change the greedy function to $\frac{\delta_v(S)}{c(v)^2}$. Greedy approaches always focus on current iteration. However the combination of several nodes is able to trigger a large cascade of adoptions. Nodes with less seeding cost are more preferable in long run. And $\frac{\delta_v(S)}{c(v)^2}$ can help to enlarge the influence of costs in the selection process. For example, we have two products, with costs and profit of c_1, c_2 and p_1, p_2 respectively. Without loss of generality, let $\frac{p_1}{c_1} \geq \frac{p_2}{c_2}$. If $c_1 \leq c_2$, obviously $\frac{p_1}{c_1^2} \geq \frac{p_2}{c_2^2}$. On the contrary, if $c_1 \geq c_2$, by applying the current objective function, we still have a chance to select product 2, which is never considered when we directly measure the ratio of $\frac{\delta_v(S)}{c(v)}$. Thus, the PMCE is more likely to choose node with less cost. Meanwhile, the second direction of modification is ignoring the impact of product costs. We iteratively pick node maximizing $\delta_v(S)$ until no more nodes can be pick without exceeding the given budget. PMCE finally returns the maximum one of these two candidate results (Algorithm 1, line 16). Neither of modifications can guarantee the performance and both candidate results can be arbitrary bad. However, we prove that at least one candidate is not that far away from the optimal solution and PMCE guarantees a constant ratio approximation.

Algorithm 1 Profit Maximization with Cost Effectiveness (PMCE)

Input: A graph $G = (V, E, \Theta, p, c)$ and budget B
Output: A set of seed nodes S^+ of cost at most B

1: $S \leftarrow \emptyset$, $curB \leftarrow 0$
2: **while** $curB \leq B$ **do**
3:   **for** $i = 1$ to q **do**
4:     $u \leftarrow argmax_{v_i \in V \setminus S} \delta_v(S)/c(v)^2$
5:   **end for**
6:   $S \leftarrow S \cup \{u\}$
7:   $curB \leftarrow curB + c(u)$
8: **end while**
   \\\\ *Construct the second candidate set* S'
9: $S' \leftarrow \emptyset$, $B' \leftarrow 0$
10: **while** $B' \leq B$ **do**
11:   **for** $i = 1$ to q **do**
12:     $u \leftarrow argmax_{v_i \in V \setminus S'} \delta_v(S')$
13:   **end for**
14:   $S' \leftarrow S' \cup \{u\}$, $B' \leftarrow B' + c(u)$
15: **end while**
16: **Return** $S^+ = argmax_{x \in \{S, S'\}} \rho(x)$

We denote $S_i = \{s_1, s_2, ..., s_i\}$ as the sequence of nodes picked by the PMCE algorithm at first i iterations, and $c(\cdot)$ is the cost function of seeding a node. Given budget B, we define the optimal solution as S^*.

Lemma 1. *For $i = 1, 2, ..., |V|$, the following lemma holds*

$$\rho(S^*) - \rho(S_{i-1}) \leq B^2 \frac{\delta_{s_i}(S_{i-1})}{c(s_i)^2}.$$

Proof. Since each node is chosen by the greedy approach, which implies, for any node $u \in S^* \setminus S_{i-1}$, we have

$$\frac{\rho(S_i) - \rho(S_{i-1})}{c(s_i)^2} \geq \frac{\rho(S_{i-1} \cup u) - \rho(S_{i-1})}{c(u)^2}.$$

Thus, we can obtain that

$$\begin{aligned} \sum_{u \in S^* \setminus S_{i-1}} \delta_u(S_{i-1}) &\leq \frac{\delta_{s_i}(S_{i-1})}{c(s_i)^2} \sum_{u \in S^* \setminus S_{i-1}} c(u)^2 \\ &\leq \frac{\delta_{s_i}(S_{i-1})}{c(s_i)^2} \left(\sum_{u \in S^* \setminus S_{i-1}} c(u) \right)^2 \\ &\leq \frac{\delta_{s_i}(S_{i-1})}{c(s_i)^2} B^2. \end{aligned}$$

Let $S^* \setminus S_{i-1} = \{x_1, x_2, ..., x_m\}$. According to the submodularity of function $\rho(\cdot)$, for $j = 1, 2..., m$, we have $\rho(S_{i-1} \cup \{x_1, x_2, ..., x_j\}) - \rho(S_{i-1} \cup \{x_1, x_2, ..., x_{j-1}\}) \leq \delta_{x_j}(S_{i-1})$. Therefore,

$$\begin{aligned}
&\rho(S^*) - \rho(S_{i-1}) \\
\leq &\rho(S^* \cup S_{i-1}) - \rho(S_{i-1}) \\
= &\rho((S^* \setminus S_{i-1}) \cup S_{i-1}) - \rho(S_{i-1}) \\
= &\rho(S_{i-1} \cup \{x_1, .., x_m\}) - \rho(S_{i-1} \cup \{x_1, .., x_{m-1}\}) \\
&\quad + \rho(S_{i-1} \cup \{x_1, .., x_{m-1}\}) - \rho(S_{i-1}) \\
\leq &\delta_{x_m}(S_{i-1}) + \rho(S_{i-1} \cup \{x_1, .., x_{m-1}\}) - \rho(S_{i-1}) \\
= &\sum_{j=1}^{m} \delta_{x_j}(S_{i-1}) \\
\leq &B^2 \frac{\delta_{s_i}(S_{i-1})}{c(s_i)^2}.
\end{aligned}$$

□

Lemma 2. *For $i = 1, 2, ..., |V|$, the following holds*

$$\rho(S_i) \geq (1 - \prod_{k=1}^{i} (1 - \frac{c(s_k)^2}{B^2}))\rho(S^*).$$

Proof. When $i = 1$, this lemma directly follows Lemma 1. Now, assuming this lemma is true for $i - 1, i \in \{2, ..|V|\}$, we prove it is also true for i.

$$\begin{aligned}
\rho(S_i) &= (\rho(S_i) - \rho(S_{i-1})) + \rho(S_{i-1}) \\
&\geq \frac{c(s_i^2)}{B^2}(\rho(S^*) - \rho(S_{i-1})) + \rho(S_{i-1}) \\
&= (1 - \frac{c(s_i)^2}{B^2})\rho(S_{i-1}) + \frac{c(s_i)^2}{B^2}\rho(S^*) \\
&\geq (1 - \frac{c(s_i)^2}{B^2})(1 - \prod_{k=1}^{i-1}(1 - \frac{c(s_k)^2}{B^2}))\rho(S^*) \\
&\quad + \frac{c(s_i)^2}{B^2}\rho(S^*) \\
&= (1 - \prod_{k=1}^{i}(1 - \frac{c(s_k)^2}{B^2}))\rho(S^*).
\end{aligned}$$

The first inequality holds when apply Lemma 1, and the second inequality follows our assumption. □

Theorem 2. *The PMCE algorithm has a guaranteed ratio of $\frac{1}{2}(1 - \frac{1}{e^2})$ for the PM^{2}A problem.*

Proof. From Lemma 2, we have

$$\begin{aligned}
\rho(S_n) &\geq (1 - \prod_{i=1}^{n}(1 - \frac{c(s_i)^2}{B^2}))\rho(S^*) \\
&\geq (1 - \prod_{i=1}^{n}(1 - (\frac{c(s_i)}{c(S_n)})^2))\rho(S^*) \\
&\geq (1 - (1 - \frac{1}{n})^{2n})\rho(S^*) \\
&\geq (1 - \frac{1}{e^2})\rho(S^*).
\end{aligned}$$

The first inequality follows Lemma 2. According to the termination rule, we have $c(S_n) = \sum_{i=1}^{n} s_i \geq B$, thus getting the second inequality. And $1 - \prod_{i=1}^{n}(1 - (\frac{c(s_i)}{c(S_n)})^2)$ achieves its minimum value of $1 - (1 - \frac{1}{n})^{2n}$ when $c(s_1) = .. = c(s_n) = c(S_n)/n$. Since the next chosen seed s_{i+1} exceeds the given overall budget, the marginal increase $\delta_{s_{i+1}}(S_n)$ is bounded by $\rho(S')$, as the final step guarantees the greater one will be returned. Henceforth, we know that at least one of the values between $\rho(S)$ and $\rho(S')$ must be greater or equal to $\frac{1}{2}(1 - \frac{1}{e^2})\rho(S^*)$. □

Theorem 3. *The PMCE algorithm has a time complexity of $O(knq(m + n))$, where $\frac{B}{\lfloor c_{max} \rfloor} \leq k \leq \frac{B}{\lfloor c_{min} \rfloor}$.*

Proof. The overall profit $\rho(S)$ of all adoptions can be computed by using a Breadth-First-Search procedure. The time taken is $O(m+n)$ which is the linear of the network size. The loops in line 3 and 11 contribute qn to the time complexity. The loops in line 2 and 10 can be $\frac{B}{\lfloor c_{max} \rfloor} \leq k \leq \frac{B}{\lfloor c_{min} \rfloor}$, and in the worst case scenario, k could be as large as $\Omega(n)$. The time taken for each of the candidate solution are the same. Thus, the total time complexity is $O(knq(m + n))$. □

B. Profit Maximization with Intelligent Selection Algorithm

The PMCE algorithm can maintain a ratio of $\frac{1}{2}(1 - e^{-2})$. However, it has two main drawbacks. First, it only looks at the current round of node status in the network, and greedily picks the one with highest marginal value of objective function, without considering any optimization on distributing overall budget across each product. Second, as shown in Theorem 3, in the worst case, it may take $O(n^2 q(m + n))$, which is not scalable for large networks. Therefore, in this subsection, we proposed another approximation algorithm Profit Maximization with Intelligent Selection (PMIS), in which we can achieve a better bound as well as running time.

The PMIS algorithm consists of two main parts. First of all, we decompose the whole network $G = (V, E, \Theta)$ to q networks $G_i = (V_i, E_i, \theta_i, c_i, p_i)$, each of which corresponds to one product. For each G_i, we want to find out the most potential seeds within budget B (line 2-11 in Algorithm 2). This step can be bounded by Theorem 1, because we separate out each product, and for each individual product, the costs and profits are uniform. After this, we are able to get a list of pairs $(S_i, \rho(S_i))$, $0 \leq i \leq q$, $0 \leq |S_i| \leq \lfloor \frac{B}{c_i} \rfloor$, where S_i is the set of nodes selected to be product i initiators, and $\rho(S_i)$ is the profit gained from product i with S_i. Without the limitation of budget, $\cup_{i=1}^{q} S_i$ can give the maximum profit. However, the cost of $\cup_{i=1}^{q} S_i$ is at most qB, which exceeds the given budget and fails to satisfy the feasibility. Therefore, after obtaining the candidate nodes from each individual network G_i, we need to distribute budget across each product over the candidate seed nodes, which is challenging.

Given products with different prices and costs, and limited budget, one may resort to solutions to the classic knapsack problem. But these methods cannot perform well here, because selecting any node from the candidate pool will influence the marginal gain of choosing the next seed, not like the static weight associated with each item in the knapsack problem. Thus it requires us to keep updating all the way to the termination condition. So we introduce the *Multiple-Choice Knapsack Problem* (MCKP), and formulate the budget distribution on candidate nodes as a MCKP.

Definition 2 (Multiple-choice Knapsack Problem (MCKP))**.** *Let q, l, B be positive integers. For each $i \in \{1, \ldots, q\}$, let*

$$C_i = \{x_{ij} : j \in \{0, \ldots, l\}\},$$

where each x_{ij} is a distinct object with profit p_{ij} and cost c_{ij}. Thus we have a problem instance $(q, l, B, \{C_i\})$. The MCKP is to pick exactly one element j_i from each class C_i, such that $\sum_{i=1}^{q} p_{ij_i}$ is maximized with $\sum_{i=1}^{q} c_{ij_i} \le B$.

Algorithm 2 Profit Maximization with Intelligent Selection (PMIS)

Input: A graph $G = (V, E, \Theta, p, c)$ with q products and budget B
Output: A set of seed nodes S of cost less or equal to B

```
1: S ← ∅
2: for i = 1 to q do
3:     G_i = (V_i, E_i, w_i, p_i, c_i)
4:     S_i ← ∅
5:     Initialize a queue Q_i
6:     while c(S_i) ≤ B do
7:         v ← argmax_{v∈V_i} δ(v)
8:         S_i ← S_i ∪ {v}
9:         Q_i.enQueue(S_i, ρ(S_i))
10:    end while
11: end for
12: Initialize a matrix P, j = 1
13: for i = 1 to q do
14:     while Q_i ≠ ∅ do
15:         p_i0 ← 0
16:         p_ij ← Q_i.deQueue()
17:         j + +
18:     end while
19: end for
20: Initialize a matrix C
21: for i = 1 to q do
22:     for j = 0 to ⌊B/c_i⌋ do
23:         c_ij ← j c_i
24:     end for
25: end for
26: {X, totalP} ← MCKP(P, C)
27: for i = 1 to q do
28:     for j = 0 to ⌊B/c_i⌋ do
29:         if x_ij = 1 then
30:             S_i ← Q_i.getElementQueue(j)
31:             S ← S ∪ S_i
32:         end if
33:     end for
34: end for
35: Return S
```

To get a corresponding MCKP instance for PM^{2}A, we first construct a profit matrix P (Algorithm 2, line 12-19), where we first have one column p_{i0} for not selecting any nodes, and then from column p_{i1} to $p_{i\lfloor \frac{B}{c_i} \rfloor}$, we assign profit achieved by having j nodes from S_i adopt product i. Similarly, we also build a cost matrix C (Algorithm 2, line 20-25). Then we are able to formulate the PM^{2}A problem as an 0-1 Integer Programming (IP) problem below.

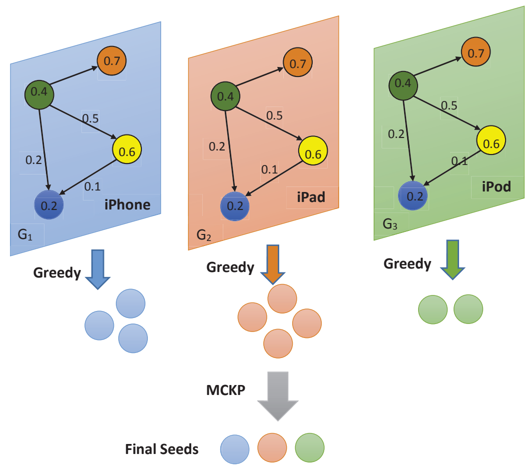


Fig. 2: An illustration of Algorithm 2.

$$\max \sum_{i=1}^{q} \sum_{j=0}^{\lfloor B/c_i \rfloor} p_{ij} x_{ij} \tag{1}$$

$$s.t. \sum_{i=1}^{q} \sum_{j=0}^{\lfloor B/c_i \rfloor} c_{ij} x_{ij} \le B \tag{2}$$

$$\sum_{j=0}^{\lfloor B/c_i \rfloor} x_{ij} \le 1, i = 1, \ldots, q \tag{3}$$

where $x_{ij} = 1$ indicates j nodes have been chosen to be initial adopters of product i. Since nodes in S_i are chosen by greedy approach, as we are going to maximize total profit, it is easy to verify that the first j nodes in order will be selected into the final solution set. Solving the IP problem (1)$\sim$(3) by CPLEX [14], we can obtain the optimal solution.

Lemma 3. *The optimal solution to MCKP instance $(q, l, B, \{C_i\})$ is an optimal solution to PM^{2}A.*

Proof. Let $\{(G_i, \sigma_i) : i \in \{1, \ldots, q\}\}$ be q distinct networks representing q different products, each equipped with submodular influence propagation. Let c_i, p_i be the cost and profit for seeding, activating a vertex with product i, respectively. Let $S_{ij}^{opt} \subset G_i$ be a seed set of size j in G_i with $\sigma(S_{ij}^{opt})$ maximized.

Next, let $l = \max_i\{\lceil \frac{B}{c_i} \rceil\}$. Define $C_i = \{S_{ij}^{opt} : j \in \{1, \ldots, l\}\}$, with $p_{ij} = p_i \cdot \sigma(S_{ij}^{opt})$, $c_{ij} = c_i \cdot j$. Any solution S to PM^2A comprises a disjoint union of S^i, the seed set consisting of only product i. The cost of S is the sum of the cost of each S^i, and must be within the budget:

$$c(S) = \sum_i c(S_i) = \sum_i c_i |S_i| \le B.$$

Hence, S corresponds to a feasible solution to MCKP, with the same profit. Furthermore, any solution to this instance of MCKP is a solution to PM^{2}A with the same cost. $\square$

Theorem 4. *PMIS produces a solution within factor $\alpha \cdot (1 - 1/e)$ of optimal, where α is the approximation ratio for MCKP and may be made arbitrarily close to* 1.

Proof. Let $S_{ij} \subset G_i$ be a seed set chosen with the greedy approximation to the maximum with j elements. So for $c = 1-1/e$, $c^{-1} \cdot \sigma(S_{ij}) \geq \sigma(S_{ij}^{opt})$ for all i, j. Let $i \in \{1, \ldots, q\}$. Let $C_i = \{x_{ij} : x_{ij}$ has profit $p_i \sigma(S_{ij}^{opt})$ and cost $jc_i\}$. The problem of picking one x_{ij} for each C_i to maximize profit while staying within budget is MCKP. Denote this problem by Ψ_1.

Define B_i similar to C_i, except that element x_{ij} has profit $p_i \sigma(S_{ij})$. Denote the MCKP on the collection $\{B_i\}$ as Ψ_2. Since $c^{-1} p_i \sigma(S_{ij}) \geq p_i \sigma(S_{ij}^{opt})$ for all i, j, we have $c^{-1} \cdot OPT(\Psi_2) \geq OPT(\Psi_1)$. Hence an α-approximation for MCKP returns on Ψ_2 a solution with profit A such that

$$c^{-1} \alpha^{-1} A \geq c^{-1} \cdot OPT(\Psi_2) \geq OPT(\Psi_1).$$

Hence, this yields a c-approximation algorithm for PM^{2}A. □

Remark: α is the approximation ratio for the MCKP problem. In our paper, by using CPLEX to solve it, we can get the optimal solution where $\alpha = 1$. Therefore, PMIS produces solution with a ratio of $1 - 1/e$.

Scalability. As Fig. 2 shows, in Algorithm 2, we separate out our network as q individual networks with single product carried on it and the selection of candidate nodes in each network is equivalent to the classic influence maximization problem. Natural greedy algorithms [4] can be applied but it suffers from scalability problem. As many existing works have been conducted [6], [13] to address this problem, we can apply these existing techniques to significantly improve the running time on selecting candidate seeds from each network. Furthermore, as the diffusion does not interfere with each other, we can run it in parallel, which allows us to market more products. On the other hand, for the second part of this algorithm, since the chosen candidate sets are usually small, the optimal solution can be quickly obtained by solving the IP problem.

IV. Empirical Evaluation

A. Experiment Setup

We use three real-world networks, which are widely used in information propagation and viral marketing analysis. The basic statistics are summarized in TABLE I, including:

NetS. Co-authorship Network in Network Science, with nodes representing authors and edges representing co-authorship [9].

BlogCatalog. BlogCatalog is a social blog directory website. Users are represented using nodes while edges describe the relationship among those bloggers [15].

Facebook. This dataset contains friendship information among New Orleans regional network on Facebook, spanning from September 2006 to January 2009 [16], where nodes stand for users and edges among them represent friendship.

Note that since all of these social networks are undirected graphs, we thereby double the number edges by reversing the orientations from the data to make every undirected edge as two directed edges.

TABLE I: Basic Information of Investigated Networks

Network	Nets	BlogCatalog	Facebook
Nodes	1,588	10,312	61,096
Edges	5,484	667,966	1,811,130
Avg. Degree	3.45	64.77	29.64

For the graphs we test on, we randomly generate products' costs and profit in three cases: (1) steady profit/cost ratio, where the profit/cost ratio among each two of them is less than 0.1; (2) monotone increasing profit/cost ratio, where the profit per unit cost goes monotone increasing; (3) monotone decreasing profit/cost ratio, where the profit per unit cost goes monotone decreasing. In addition, for node $v \in V$'s threshold vector Θ_v, we draw each of the thresholds θ_v^i from the range [0,1] for $1 \leq i \leq q$ uniformly at random. For the influence weights associated with edges, we firstly generated them uniformly at random from the range $[0, 1]$, and then normalize all weights of the incoming edges of a node v to make their summation equal to 1 [6].

TABLE II: Basic Information of Products

Network	Nets	BlogCatalog	Facebook
Products	P_1, P_2, P_3	P_1, P_2, P_3	P_1, P_2, P_3
Profit	0.43,0.51,0.88	0.41,0.55,0.44	0.44,0.30,0.05
Cost	0.33,0.42,0.72	0.97,0.57,0.09	0.17,0.67,0.64
Ratio	1.28,1.22,1.23	0.42,0.96,4.89	2.59,0.45,0.08

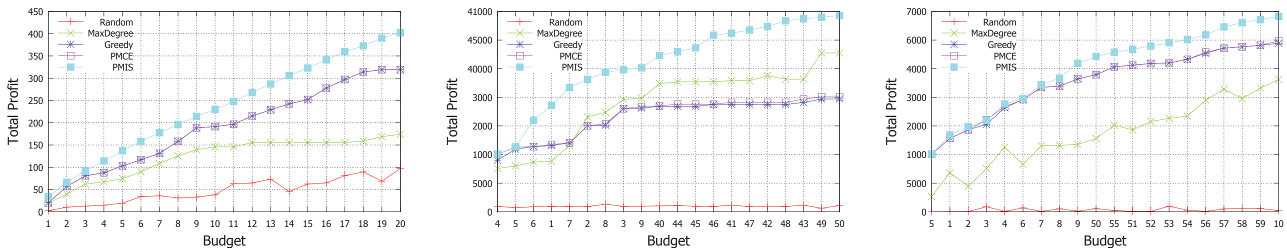
B. Algorithms Compared

To evaluate the performance, we compare our algorithm with several other widely used algorithms. As our goal is to maximize the overall profit, for fairness of the evaluation, we adapt each of them to make the objective function be consistent with our target:

- Random: Randomly select nodes within budget B and randomly seed node from three products.
- MaxDegree: Select nodes with highest degree within budget B and assign all products to them in order.
- Greedy: Select node with largest marginal gain of $\delta_v(S)/c(v)$ in each iteration.
- PMCE: Algorithm 1 introduced in our paper.
- PMIS: Algorithm 2 proposed in this paper.

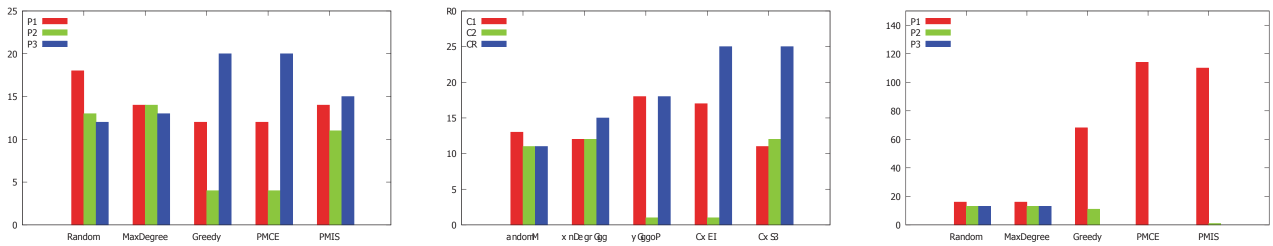
C. Experiment Results

Profit Achieved within Budget. We begin with comparing the total profit gained by different heuristics with budget ranges from 1 to 20. We evaluate them based on the total profit of all activated nodes across all products. Since different budget results in different combinations of seed selection, we run all algorithms for budget from 1 to 20 separately. Fig. 3 shows the profit achieved for each budget with three products listed in TABLE II.

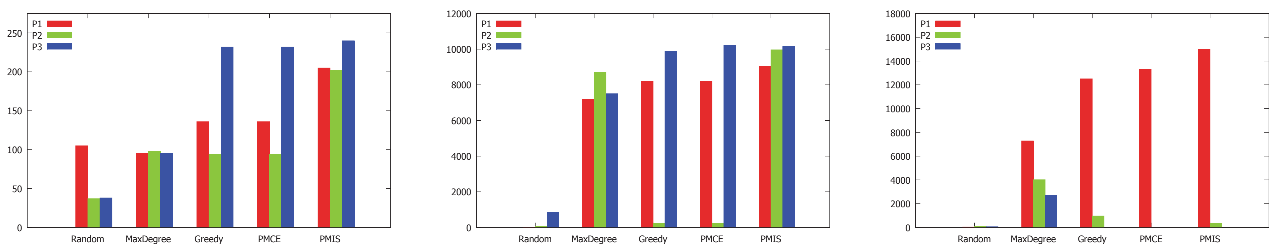


(a) Total profit in Nets (b) Total profit in BlogCatalog (c) Total profit in Facebook

Fig. 3: Total profit achieved by various algorithm with budget B from 1 to 20.



(a) Seeds adoptions distribution in Nets (b) Seeds adoptions distribution in BlogCatalog (c) Seeds adoptions distribution in Facebook

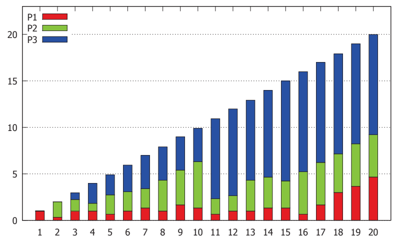


(d) All adoptions distribution in Nets (e) All adoptions distribution in BlogCatalog (f) All adoptions distribution in Facebook

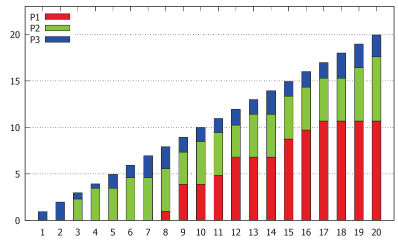
Fig. 4: Distribution of budget, seeds adoptions and all adoptions.

First, we can see that with the increment of budget, the overall profit achieved is steadily increasing. It is shown that the seeds set selected by our PMIS is of high quality, and even beats the result of two greedy algorithms. For Random method, the total profit has been oscillating close to zero. MaxDegree method performs much better than Random, but does not perform as good as our algorithms . For example, in BlogCatalog, the overall profit gained from 3603 all the way to 7887 with budget 1 to 20 in Greedy approach, while the PMIS reaches 13723 finally, which is almost two times larger than Greedy. One interesting phenomenon is shown in Facebook, where the MaxDegree sometimes drops down with increment of budget. We investigate into it and find out that different budget leads to totally different combination of seeds selection. For example, with budget $B = 2$, three nodes have been activated with one of them adopts three products and two of them adopt one product (with cost $c = 0.17$, and profit $p = 0.44$), since we cannot seed the second node with either of the rest two products. When the budget increases to $B = 3$, we seed two nodes with three products. However, the first assignment with less budget results in more profit compared to the second one with more budget.

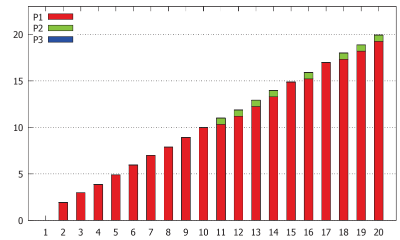
Distribution of Adoptions. We are interested in the constitution of seeds and how is the adoption behavior look like among all adopters in three networks. Fig. 4 presents the distribution of adoptions among seeds and all activated users when $B = 20$. For different assignment of costs and profits of three products, we can clearly see that the adoption behavior is totally different from each other. With steady relationship of costs and profits, we have all products been adopted in Nets for each algorithm, as presented in Fig. 4(a). However, in Facebook, there is no users adopt product 2 and 3 when PMCE approach is applied. In addition, comparing Fig. 4(a)-(c) to Fig. 4(d)-(f), we can see that the more initial adopters on one product, the more total adoptions will be achieved. Furthermore, Greedy, PMCE and PMIS tend to select more seeds with high profit/cost ratio, which has been fully reflected in Facebook, where *PMCE* lets all selected seeds adopt product P_1, and this product is also preferable to Greedy as well as $PMIS$. The notable difference between greedy approaches and PMIS is that, since two greedy algorithms focus on the gain of current iteration, they favorite products with high profit/cost ratio and low costs, and fail to look



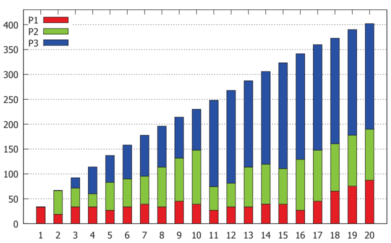
(a) Budget distribution in Nets



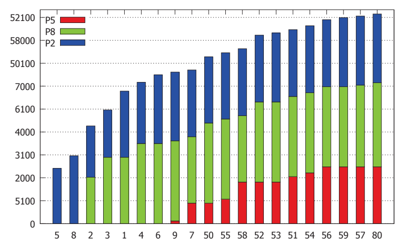
(b) Budget distribution in BlogCatalog



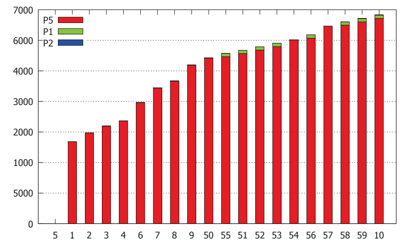
(c) Budget distribution in Facebook



(d) Profit distribution in Nets



(e) Profit distribution in BlogCatalog



(f) Profit distribution in Facebook

Fig. 5: Distribution of budget and total profit of all products.

forward for the combination effect. However, PMIS can better balance the costs and profits in long run, and come up a with better combination of seeds. For example, in BlogCatalog, PMCE has 17 and 25 initial activated users for product P_1 and P_3, while it only chooses 1 node for product P_2. However, the PMIS chooses 11 and 25 users for product P_1 and P_3 respectively, and 12 users for product P_2, which leads to almost two times overall profit than PMCE shown by Fig. 3(b).

Distribution of Budget and Profit. After discussing the distribution of adoptions among seeds and all activated users, we see that PMIS outperforms other heuristics in producing seeds. We go one step further to investigate into it, where we want to see with the increment of budget, how it works for distributing the overall budget across three products with PMIS. Fig. 5 demonstrates the distribution of budget and total profit achieved for each product. From Fig. 3, we see that in the initial stage, Greedy and PMCE almost achieve the same total profit as PMIS. However, when the budget increases, the difference is more and more significant between those two greedy approaches and PMIS. By looking into the result, we find an interesting phenomenon: keeping spending more money on one profitable product does not necessarily benefit the overall profit. For example, product P_3 is the most profitable in BlogCatalog, therefore, with limited budget in the beginning, every penny is spent on it. With more money spent on it, the profit achieved is going up sharply. However, when it reaches 4304 with total expense 1.32, it increases very slowly. If we keep spending money on it until using up the total budget by greedy approach for this single product, 4572 will be obtained finally, where 93% budget contributes for 5% increment for the profit. Therefore, PMIS can effectively avoid sticking with one product, and optimize the spending on the situation as a whole. As a final result, it yields much more profit.

V. Discussion

We are mainly studying the profit maximization problem of multiple adoptions in this work. However, consider personal budget and similar functionalities of some products, people may need to make a decision and adopt one of exiting products. For example, when both of iPhone 6 and iPhone 6 plus come into market, few people will purchase two of them, most people select one to adopt.

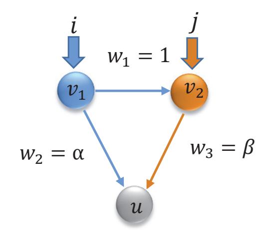


Fig. 6: A counter example for the monontonicity.

In PM^{2}A problem, we have proved that the profit function $\rho(\cdot)$ is monotone increasing and submodular in Theorem 1. However, if every individual is limited to make one adoption, the profit function $\rho(\cdot)$ is no longer monotone increasing nor submodular. We show this by the example in Fig 6. There are three nodes v_1, v_2 and u, the weights are assigned as follows, $w_{v_1,v_2} = 1, w_{v_1,u} = \alpha, w_{v_1,u} = \beta$, and $\alpha, \beta \in [0, 1]$, and $\alpha + \beta \leq 1$. Now, consider $S = \{v_1\}$, and seed v_1 with product i, then the expected total profit is $\rho(S) = [2 + \alpha + \beta]p_i$. Then, let $T = \{v_1, v_2\}$, in which we seed v_1 with product i, and v_2 with product j. as a result, the expected profit of T is

$$\rho(T) = [1+\alpha(1-\beta)+\alpha\beta(\frac{\alpha}{\alpha+\beta})]p_i + [1+\beta(1-\alpha)+\alpha\beta(\frac{\beta}{\alpha+\beta})]p_j.$$

Then let $\alpha = \beta = 0.5$, and $p_i = p_j = 1$, we can obtain that $\rho(S) = 3$, and $\rho(T) = 2.85 \leq \rho(S)$, where $S \subset T$.

Henceforth, in this scenario, maximizing overall profit is more challenging since products themselves compete with each other. We will investigate into this case in the future work.

VI. Related Works

The impact of social influence has been demonstrated in various applications, such as viral marketing [4], [17], [18], misinformation blocking [12], recommendation system [19] and so on. Given a graph, how to detect the seeds to serve with the corresponding ultimate goal is a major research topic. Kempe et al. [4] first formally formulated the influence maximization problem and proposed a greedy algorithm with a ratio of $(1 - 1/e)$. Later on, numerous works about improving the scalability and efficiency have been done [6], [13]. In addition, researchers also take into account real world scenarios to adapt diffusion models to handle more complicated problems. Bhagat et al. [17] adapted the Linear Threshold model by defining an objective function that explicitly differentiate adoption from influence among users. In [18], the authors took into account user opinions and focusing on maximizing the positive influence instead of overall influence, in which they proposed an opinion cascade model to incorporate with user opinions. In our paper, we also adopt the Linear Threshold model, and extend it to handle multiple diffusion of different products.

With respect to adoptions and products costs and profit, we formulate the profit maximization of multiple adoptions problem. There are some recent works investigate multiple diffusion problem. [11] discusses competitive influence maximization for two competitors and give solution to an opponent's strategy. A more recent work [20] consider the budgeted influence maximization problem. However, they are mainly focusing on one diffusion of one particular kind of product, which is different from our work, since we are studying the multiple diffusion of different products. Another related work is [21], where authors also study the multiple products problem, but the diffusion corresponding to each product is separate with each other because they have different budgets. In our paper, all diffusions are related with each other as they have an overall budget, and this makes our problem more challenging.

VII. Conclusion

Traditional influence maximization problem focuses on studying single diffusion in the network, and aims at maximizing the number of activated people of a product or service. However, considering that companies usually produce various products for supplying customers with different demands. Therefore, how to wisely distribute a limited budget across multiple products to achieve the maximum profit is more meaningful in designing commercial activities.

In this paper, we investigate into the problem of profit maximization with multiple adoptions. We propose two effective algorithms to find seeding set with theoretical guaranteed performance. Furthermore, we conduct extensive experiments on three real world datasets. Results from the experiments show that our proposed algorithms can output seeds with high quality and achieve more overall profit than other heuristics.

VIII. Acknowledgment

This work is partially supported by NSF CCF-1422116.

References

[1] "The top 500 sites on the web," 2014, http://www.alexa.com/topsites.
[2] "Rokenbok," 2014, https://www.youtube.com/yt/advertise/success-stories.html.
[3] "State bicycle," 2014, https://www.facebook.com/business/success/state-bicycle.
[4] D. Kempe, J. M. Kleinberg, and V. Tardos, "Maximizing the spread of influence through a social network," in *KDD*, 2003, pp. 137–146.
[5] W. Chen, A. Collins, R. Cummings, T. Ke, Z. Liu, D. Rincn, X. Sun, Y. Wang, W. Wei, and Y. Yuan, "Influence maximization in social networks when negative opinions may emerge and propagate," in *SDM*, 2011, pp. 379–390.
[6] W. Chen, Y. Yuan, and L. Zhang, "Scalable influence maximization in social networks under the linear threshold model," in *ICDM*, 2010.
[7] T. N. Dinh, H. Zhang, D. T. Nguyen, and M. T. Thai, "Cost-effective viral marketing for time-critical campaigns in large-scale social networks," *IEEE/ ACM Transactions on Networking (ToN)*, 2013.
[8] H. Zhang, S. Mishra, and M. T. Thai, "Recent advances in information diffusion and influence maximization in complex social networks," in *Opportunistic Mobile Social Networks*, J. Wu and Y. Wang, Eds. CRC Press, Taylor and Francis Group, 2014.
[9] D. T. Nguyen, H. Zhang, S. Das, M. T. Thai, and T. N. Dinh, "Least cost influence in multiplex social networks: Model representation and analysis," in *ICDM*, 2013.
[10] N. Pathak, A. Banerjee, and J. Srivastava, "A generalized linear threshold model for multiple cascades," in *ICDM*, 2010, pp. 965–970.
[11] S. Bharathi, D. Kempe, and M. Salek, "Competitive influence maximization in social networks," in *WINE*, 2007, pp. 306–311.
[12] C. Budak, D. Agrawal, and A. E. Abbadi, "Limiting the spread of misinformation in social networks," in *WWW*, 2011, pp. 665–674.
[13] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. S. Glance, "Cost-effective outbreak detection in networks," in *KDD*, 2007.
[14] I. CPLEX, "High-performance software for mathematical programming and optimization," 2005.
[15] R. Zafarani and H. Liu, "Social computing data repository at ASU," 2009. [Online]. Available: http://socialcomputing.asu.edu
[16] B. Viswanath, A. Mislove, M. Cha, and K. P. Gummadi, "On the evolution of user interaction in facebook," in *WOSN*, 2009, pp. 37–42.
[17] S. Bhagat, A. Goyal, and L. V. Lakshmanan, "Maximizing product adoption in social networks," in *WSDM*, 2012, pp. 603–612.
[18] H. Zhang, T. N. Dinh, and M. T. Thai, "Maximizing the spread of positive influence in online social networks," in *ICDCS*, 2013, pp. 317–326.
[19] J. Leskovec, A. Singh, and J. Kleinberg, "Patterns of influence in a recommendation network," in *Advances in Knowledge Discovery and Data Mining*. Springer, 2006, pp. 380–389.
[20] H. Nguyen and R. Zheng, "On budgeted influence maximization in social networks," *Selected Areas in Communications, IEEE Journal on*, vol. 31, no. 6, pp. 1084–1094, 2013.
[21] M. F. B. Nan Du, Yingyu Liang and L. Song, "Budgeted influence maximization for multiple products," *arXiv:1312.2164*.