Profit Maximization for Viral Marketing in Online Social Networks

Jing TANG
Interdisciplinary Graduate School
Nanyang Technological University
Email: tang0311@ntu.edu.sg

Xueyan TANG
School of Comp. Sci. & Eng.
Nanyang Technological University
Email: asxytang@ntu.edu.sg

Junsong YUAN
School of EEE
Nanyang Technological University
Email: jsyuan@ntu.edu.sg

***Abstract*—Information can be disseminated widely and rapidly through Online Social Networks (OSNs) with "word-of-mouth" effects. Viral marketing is such a typical application in which new products or commercial activities are advertised by some seed users in OSNs to other users in a cascading manner. The budget allocation for seed selection reflects a tradeoff between the expense and reward of viral marketing. In this paper, we define a general profit metric that naturally combines the benefit of influence spread with the cost of seed selection in viral marketing to eliminate the need for presetting the budget for seed selection. We carry out a comprehensive study on finding a set of seed nodes to maximize the profit of viral marketing. We show that the profit metric is significantly different from the influence metric in that it is no longer monotone. As a result, from the computability perspective, the problem of profit maximization is much more challenging than that of influence maximization. We develop new seed selection algorithms for profit maximization with strong approximation guarantees. Experimental evaluations with real OSN datasets demonstrate the effectiveness of our algorithms.**

I. Introduction

Online Social Networks (OSNs), such as Facebook, Twitter, Flickr, Google+, and LinkedIn, are heavily used today in terms of not only the number of users but also their time consumption. Information can be disseminated widely and rapidly through OSNs with "word-of-mouth" effects. Leveraging OSNs as the medium for information spread has been increasingly adopted in many areas. Viral marketing is such a typical application in which new products or commercial activities are advertised by some influential users in the OSN to other users in a cascading manner [8].

A large amount of recent work [6], [7], [15], [16], [17], [22], [25], [26], [27], [29] has been focusing on *influence maximization* in viral marketing, which targets at selecting a set of initial seed nodes in the OSN to spread the influence as widely as possible. The seminal work by Kempe *et al.* [16] formulated the influence maximization problem with two basic diffusion models, namely the *Independent Cascade* (IC) and *Linear Threshold* (LT) models. Although finding the optimal seed set is NP-hard [16], a simple greedy algorithm has a $(1-1/e)$-approximation guarantee due to the submodularity and monotone properties of the influence spread under these models [21]. Many follow-up studies have concentrated on efficient implementation of the algorithm for large-scale OSNs [6], [7], [15], [17], [22], [25], [26], [27], [29].

All the above work has assumed a fixed and pre-determined budget for seed selection. In essence, the cost of seed selection is the price to pay for viral marketing (e.g., providing the selected users with free samples or other incentives). The influence spread, on the other hand, is the reward of viral marketing, which can potentially be translated into growth in the adoptions of products. Thus, the budget for seed selection reflects a tradeoff between the expense and reward of viral marketing. If the budget is set too low, it may not produce the desired extent of influence spread to fully exploit the potential of viral marketing in boosting sales and revenues. In contrast, if the budget is set too high, the benefit of the influence spread generated may not pay off the expense. Therefore, the budget allocation for seed selection is a tough problem by itself. A cost-effective budget setting should strike a balance between the expense and reward of viral marketing.

Economic-wise, a common goal for companies conducting viral marketing is to maximize the profit return, which can be defined as the reward less the expense. Thus, in this paper, we define a general profit metric that naturally combines the benefit of influence spread with the cost of seed selection to eliminate the need for presetting the budget for seed selection. We carry out a comprehensive study on finding a set of seed nodes to optimize the profit of viral marketing. We show that the profit metric is significantly different from the influence metric in that it is no longer monotone. Therefore, from the computability perspective, the problem of profit maximization is much more challenging than that of influence maximization. We show that applying simple hill-climbing algorithms to the profit maximization problem would not provide any strong theoretical guarantee on the seed set selected. Observing that seed selection for profit maximization is an unconstrained submodular maximization problem, we develop new seed selection algorithms based on the ideas of the double greedy algorithms by Buchbinder *et al.* [4]. The original double greedy algorithms have a serious limitation for our profit maximization problem: they rely on a rather strict condition for offering non-trivial approximation guarantees which is not realistic in viral marketing. We propose several new techniques to address this limitation.

Our contributions are summarized as follows.

- We define a general problem of profit maximization for viral marketing in OSNs. We show that the profit metric is submodular but not always monotone.

- We construct a greedy hill-climbing algorithm and show that such an intuitive greedy algorithm does not have any bounded approximation factor for profit maximization.
- We present double greedy algorithms to optimize the profit. We develop an iterative pruning technique to provide good warm-starts and expand the applicability of the strong approximation guarantees for the double greedy algorithms.
- We derive several online bounds on the quality of the solution obtained by any algorithm, which can be used to evaluate the practical performance of the algorithm on any specific problem instance.
- We conduct extensive experiments with several real OSN datasets. The results demonstrate the effectiveness of our profit maximization algorithms.

The rest of this paper is organized as follows. Section II reviews the related work. Section III defines the profit maximization problem. Section IV elaborates our algorithm design. Section V presents the experimental study. Finally, Section VI concludes the paper.

II. Related Work

Influence Maximization. Kempe *et al.* [16] formulated influence maximization as a discrete optimization problem, which targets at finding a fixed-size set of seed nodes to produce the largest influence spread. They derived a $(1-1/e)$-approximation greedy algorithm. Since then, there has been considerable research on improving the efficiency of the greedy algorithm by avoiding unnecessary influence estimation for certain seed sets [7], [17], using heuristics to trade the accuracy of influence estimation for computational efficiency [6], [15], or optimizing the Monte-Carlo simulations for influence estimation [22], [24], [26], [27]. In addition, some recent work [11], [19], [28] studied the seed minimization problem that focuses on minimizing the seed set size (or cost) for achieving a given amount of influence spread. Different from the above studies, we aim to maximize the profit that accounts for both the benefit of influence spread and the cost of seed selection in viral marketing.

Viral Marketing. Viral marketing in OSNs has emerged as a new way to promote the sales of products. Domingos *et al.* [8] were the first to exploit social influence for marketing optimization by modeling social networks as Markov random fields. Li *et al.* [18] modeled the product advertisement in large-scale OSNs through local mean field analysis. The model is designed to compute the expected proportion of users who would eventually buy the product, which may indirectly guide the advertisement to improve the profit. However, no specific strategy was given to maximize the profit. Hartline *et al.* [12] aimed to find the optimal marketing strategies by controlling the price and the order of sales to different customers to improve the profit. Aslay *et al.* [2] studied strategic allocation of ads to users by leveraging social influence and the propensity of ads for viral propagation. The objective is to help OSN hosts match their ad services with advertiser budgets as close as possible. Two recent studies [20], [30] also focused on finding the pricing strategies to optimize the profit return of viral marketing and they adopted a simple hill-climbing heuristic to select initial seed nodes. We show that the simple hill-climbing approach for seed selection lacks bounded approximation guarantees (Section IV-A) and may give poor performance in practice (Section V-B). We propose new algorithms to address the profit maximization problem with strong theoretical guarantees.

Unconstrained Submodular Maximization. The influence functions under typical diffusion models are submodular and monotone [16]. However, the profit function that we define is submodular but not necessarily monotone. Feige *et al.* [10] developed local search algorithms for approximately maximizing non-monotone submodular functions. Recently, Buchbinder *et al.* [4] proposed double greedy algorithms that have much lower computational complexities than those in [10] and improve the approximation guarantees to match the known hardness result of the problem. We make use of these state-of-the-art algorithms for profit maximization (Section IV-B). To avoid exploring the entire ground set, we propose a novel iterative method to prune the search space for maximizing a submodular function and apply it to our profit maximization problem to improve the efficiency of its solutions and expand the applicability of their approximation guarantees (Section IV-C). Iyer, Jegelka and Bilmes [13], [14] gave two modular upper bounds on submodular functions and made use of them for submodular minimization. Inspired by these studies, we establish several online upper bounds on the optimal solution to the submodular maximization problem for benchmarking the performance of profit optimization algorithms (Section IV-D).

III. Problem Formulation

A. Preliminaries

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a directed graph modeling an OSN, where the nodes $\mathcal{V}$ represent users and the edges $\mathcal{E}$ represent the connections among users (e.g., friendships on Facebook, followships on Twitter). For each directed edge $(u, v) \in \mathcal{E}$, we refer to v as a *neighbor* of u, and refer to u as an *inverse neighbor* of v.

There are many influence propagation models in social networks. While our problem formulation and solutions are general and not restricted to a specific influence propagation model, to facilitate exposition, we shall discuss our examples and conduct experimental evaluation using the Independent Cascade (IC) model — a representative and most widely-studied model for influence propagation [6], [7], [15], [16], [17], [22], [25], [26], [27], [29]. In the IC model, a propagation probability $p_{u,v}$ is associated with each edge (u, v), representing the probability for v to be activated by u through their connection. Let $\mathcal{N}_u$ denote the set of node u's neighbors, i.e., $\mathcal{N}_u = \{v : v \in \mathcal{V}, (u, v) \in \mathcal{E}\}$. Given a set of seed nodes $\mathcal{S}$, the IC diffusion process proceeds as follows. Initially, the seed nodes $\mathcal{S}$ are activated, while all the other nodes are not activated. When a node u first becomes activated, it has a single chance to activate its neighbors who are not yet activated. For each such neighbor $v \in \mathcal{N}_u$, v would become activated with probability $p_{u,v}$. This process repeats until no more node can be activated. The *influence spread* of the seed set $\mathcal{S}$, denoted by $\sigma(\mathcal{S})$, is the expected number of nodes activated by the above diffusion process.

B. The Profit Maximization Problem

Most previous studies have focused on maximizing the influence spread with a fixed-size seed set [6], [7], [15], [16], [22], [24], [26], [27]. That is, the problem is to find a given number of k nodes to produce the maximum influence spread, i.e., to maximize $\sigma(\mathcal{S})$ subject to $|\mathcal{S}| = k$.

Unlike existing studies, we do not assume any pre-determined number of seeds to select. As discussed, the influence spread is the benefit gained by viral marketing and the cost of seed selection is the price to pay for viral marketing. Suppose that each node $v \in \mathcal{V}$ is associated with a cost $c(v)$ for seed selection. Then, we naturally define a unified *profit* metric as the benefit of influence spread less the cost of seed selection. For simplicity, in our discussion, we shall assume that activating each node in the OSN offers the same benefit and simply represent the benefit of influence spread by the number of nodes activated. As a result, the profit of a seed set $\mathcal{S}$, denoted by $\phi(\mathcal{S})$, is given by the influence spread less the cost of seed selection, i.e.,

$$\phi(\mathcal{S}) = \sigma(\mathcal{S}) - c(\mathcal{S}),$$

where $c(\mathcal{S}) = \sum_{v \in \mathcal{S}} c(v)$ is the total cost of all the seed nodes selected. Our analysis and algorithms can also handle different weights associated with the nodes to model their respective benefits if activated (e.g., users have different tendencies to buy certain products due to their genders, ages, or occupations).

Our goal is to find a seed set $\mathcal{S}$ to maximize the profit $\phi(\mathcal{S})$. First, we study the submodularity of the profit function. To simplify the notations, we define

$$\sigma(v|\mathcal{S}) \triangleq \sigma(\mathcal{S} \cup \{v\}) - \sigma(\mathcal{S})$$

as the *marginal influence gain* of adding a new seed node $v \in \mathcal{V}$ into a seed set $\mathcal{S} \subseteq \mathcal{V}$ and define

$$\phi(v|\mathcal{S}) \triangleq \phi(\mathcal{S} \cup \{v\}) - \phi(\mathcal{S})$$

as the *marginal profit gain* of adding v into a seed set $\mathcal{S}$.

Proposition 1: The profit function $\phi(\cdot)$ is submodular if the influence function $\sigma(\cdot)$ is submodular.

Proof: If $\sigma(\cdot)$ is submodular, for any two seed sets $\mathcal{S}$ and $\mathcal{T}$ where $\mathcal{S} \subseteq \mathcal{T}$ and any node $v \notin \mathcal{T}$, it holds that

$$\sigma(v|\mathcal{S}) \geq \sigma(v|\mathcal{T}).$$

Therefore, we have

$$\begin{aligned} \phi(v|\mathcal{S}) &= \sigma(v|\mathcal{S}) - c(v) \\ &\geq \sigma(v|\mathcal{T}) - c(v) \\ &= \phi(v|\mathcal{T}), \end{aligned}$$

which implies that $\phi(\cdot)$ is also submodular. ■

Kempe *et al.* [16] has proved that the influence function $\sigma(\cdot)$ is submodular under the IC model. Thus, the profit function $\phi(\cdot)$ is also submodular under the IC model. Though both functions are submodular, it should be noted that the profit $\phi(\cdot)$ is significantly different from the influence spread $\sigma(\cdot)$ in that $\phi(\cdot)$ may no longer be monotone because the marginal profit gain by adding a new seed, i.e., $\phi(v|\mathcal{S}) = \sigma(v|\mathcal{S}) - c(v)$, can be negative. As shall be shown soon, this makes the seed selection for profit maximization much more challenging than that for influence maximization. Selecting seed nodes to maximize the profit becomes an unconstrained submodular maximization problem [4], [10].

IV. Seed Selection Algorithms

In this section, we first study an intuitive greedy algorithm and show that it does not provide any bounded approximation guarantee. To provide strong approximation guarantees, we then borrow ideas from the double greedy algorithms [4], and develop new methods for profit maximization.

A. Simple Greedy Heuristic

Intuitively, one may argue that the profit maximization problem can be easily solved by a straightforward approach that runs an influence maximization algorithm for every possible seed set size (from 1 to $|\mathcal{V}|$) and then chooses the best solution. But unfortunately, this would not work when the nodes have different costs for seed selection because most existing influence maximization algorithms [6], [7], [15], [16], [22], [24], [26], [27] do not differentiate the nodes by their costs in the seed selection. If a top influential node has a quite high cost (e.g., a popular user may require more incentives to be recruited as a seed), the total benefit of the seed set selected can be very low or even negative.

Rather than applying the influence maximization algorithms directly, we construct a simple greedy hill-climbing algorithm to optimize the profit, similar to that proposed by Kempe *et al.* [16] for influence maximization. Algorithm 1 describes the greedy heuristic. It starts with an empty seed set $\mathcal{S} = \emptyset$. In each iteration, if the largest marginal profit gain $\phi(v|\mathcal{S})$ by choosing a new seed from the non-seed nodes $\mathcal{V} \setminus \mathcal{S}$ is positive, the greedy heuristic adds the corresponding node to $\mathcal{S}$. Otherwise, it implies that the profit cannot be further increased by adding any new seed, so the algorithm stops and returns the seed set $\mathcal{S}$. This simple greedy algorithm shares the same spirit with the hill-climbing heuristic adopted by [20] and [30].

Algorithm 1: $\mathit{SimpleGreedy}(\mathcal{G}, \phi)$

```
initialize S ← ∅;
while True do
    find u ← arg max_{v∈V\S} {φ(v|S)};
    if φ(u|S) ≤ 0 then
        break;
    S ← S ∪ {u};
return S;
```

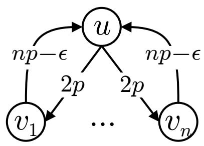


Figure 1. A simple hill-climbing algorithm fails to achieve any bounded approximation factor.

Unfortunately, the above simple greedy algorithm does not have any bounded approximation factor for profit maximization because the profit function is submodular but not monotone. This is true even if all nodes have the same unit costs for seed selection. Figure 1 shows an example social network with $n+1$ nodes ($n \geq 2$) and $2n$ edges. The propagation probabilities are given by $p_{u,v_i} = 2p$ and $p_{v_i,u} = np - \epsilon$ for each $1 \leq i \leq n$, where $\epsilon > 0$. When node u is chosen as the only seed, the probability for each node v_i to be activated is $2p$, so the profit $\phi(\{u\}) = 1 + 2np - 1 = 2np$. For each node v_i, when v_i is chosen as the only seed, the probability for node u to be activated is $np - \epsilon$ and only when u is activated, each remaining node v_j ($j \neq i$) can be activated with probability $2p$. Hence, the profit $\phi(\{v_i\}) = 1 + (np-\epsilon)\cdot\big(1+2(n-1)p\big) - 1 = (np-\epsilon)\cdot\big(1+2(n-1)p\big)$. When $p < \frac{1}{2(n-1)}$, we have $1 + 2(n-1)p < 2$ and thus, $\phi(\{v_i\}) < \phi(\{u\})$. If the simple greedy algorithm is applied, u would be the first seed selected. Furthermore, for any node v_i, $\phi(\{u, v_i\}) = 2 + 2(n-1)p - 2 = 2(n-1)p$, which means $\phi(u|\{v_i\}) = 2(n-1)p - 2np = -2p < 0$. Therefore, the greedy algorithm would stop after selecting u and return $\mathcal{S} = \{u\}$ with a profit of $\phi(\{u\}) = 2np$. On the other hand, when nodes $v_1, v_2, \ldots, v_n$ are all chosen as seeds, the probability for node u to be activated is $1-(1-np+\epsilon)^n$. As a result, $\phi(\mathcal{V}\setminus\{u\}) = n+1-(1-np+\epsilon)^n - n = 1-(1-np+\epsilon)^n$. Let $p = \frac{1}{n^2} < \frac{1}{2(n-1)}$ and $\epsilon = \frac{1}{4n^2}$. Then, we have $\phi(\{u\}) = \frac{2}{n}$ and $\phi(\mathcal{V}\setminus\{u\}) = 1-(1-\frac{1}{n}+\frac{1}{4n^2})^n = 1-(1-\frac{1}{2n})^{2n}$. When $n \to \infty$, we have $\frac{2}{n} \to 0$ and $1-(1-\frac{1}{2n})^{2n} \to 1-\frac{1}{e}$ which is a positive constant. Thus, the simple greedy algorithm can perform arbitrarily worse than the optimal solution and does not have any bounded approximation factor.

We further remark that selecting seeds based on seed minimization algorithms [11], [19], [28] for achieving a given amount of influence spread would not be able to provide strong approximation guarantees for profit maximization either. As analyzed in [11], [28], for any $\epsilon > 0$, the seed minimization problem cannot be approximated within a factor of $(1-\epsilon)\ln|\mathcal{V}|$ unless NP has $n^{O(\log\log n)}$-time deterministic algorithms. Therefore, the seed minimization algorithms do not have any constant approximation factor by themselves. Thus, we do not adapt seed minimization algorithms to our profit maximization problem.

B. Double Greedy Algorithms

Buchbinder *et al.* [4] proposed double greedy algorithms to address the unconstrained submodular maximization problem with strong approximation guarantees for non-negative submodular functions. A deterministic double greedy algorithm yields (1/3)-approximation, while a randomized double greedy algorithm yields (1/2)-approximation. Algorithm 2 describes the ideas of these algorithms in our context of profit maximization. The algorithms start with an empty set $\mathcal{S}$ and a set $\mathcal{T}$ initialized with the entire node set of the social network. They iterate through all the nodes in the network in an arbitrary order to decide whether or not to include them in $\mathcal{S}$ and $\mathcal{T}$. When the algorithms complete, it must hold that $\mathcal{S} = \mathcal{T}$ and this is the seed set selected. The decision for each node u is made based on the marginal profit gain of adding u into $\mathcal{S}$ (i.e., $\phi(\mathcal{S}\cup\{u\}) - \phi(\mathcal{S}) = \phi(u|\mathcal{S})$) and the marginal profit gain of removing u from $\mathcal{T}$ (i.e., $\phi(\mathcal{T}\setminus\{u\}) - \phi(\mathcal{T}) = -\phi(u|\mathcal{T}\setminus\{u\})$). In the deterministic approach, each node u joins $\mathcal{S}$ if it generates higher marginal profit gain than that of quitting from $\mathcal{T}$ and vice versa (lines 5–8). In the randomized approach, each node u is added to $\mathcal{S}$ with probability $\phi(u|\mathcal{S})/\big(\phi(u|\mathcal{S}) - \phi(u|\mathcal{T}\setminus\{u\})\big)$, and is removed from $\mathcal{T}$ with probability $-\phi(u|\mathcal{T}\setminus\{u\})/\big(\phi(u|\mathcal{S}) - \phi(u|\mathcal{T}\setminus\{u\})\big)$.

Algorithm 2: $\boldsymbol{DeterministicDoubleGreedy}(\mathcal{G}, \phi)$[4]

1 start with $\mathcal{S} \leftarrow \emptyset$, $\mathcal{T} \leftarrow \mathcal{V}$;
2 **for** *each node* $u \in \mathcal{V}$ **do**
3 $\quad r^+ \leftarrow \phi(u|\mathcal{S})$;
4 $\quad r^- \leftarrow -\phi(u|\mathcal{T}\setminus\{u\})$;
5 $\quad$**if** $r^+ \geq r^-$ **then**
6 $\quad\quad \mathcal{S} \leftarrow \mathcal{S}\cup\{u\}$;
7 $\quad$**else**
8 $\quad\quad \mathcal{T} \leftarrow \mathcal{T}\setminus\{u\}$;
9 **return** $\mathcal{S}$ ($=\mathcal{T}$);
// For randomized double greedy, change the condition of line 5 to $U(0,1) \leq r^+/(r^+ + r^-)$, where $U(0,1)$ is a uniformly distributed number between 0 and 1, and $r^+/(r^+ + r^-) = 1$ if $r^+ + r^- = 0$.

In general, in the double greedy algorithms, if we initialize $\mathcal{S}$ with $\mathcal{S}_0$ and $\mathcal{T}$ with $\mathcal{T}_0$ where $\emptyset \subseteq \mathcal{S}_0 \subseteq \mathcal{T}_0 \subseteq \mathcal{V}$ (line 1 in Algorithm 2) and only check the nodes in $\mathcal{T}_0 \setminus \mathcal{S}_0$ to decide whether or not to include them in $\mathcal{S}$ and $\mathcal{T}$ (line 2), we have the following proposition according to [4].[1]

Proposition 2: Let $\mathcal{S}^*$ denote the optimal solution such that $\phi(\mathcal{S}^*) = \max_{\mathcal{S}\subseteq\mathcal{V}} \phi(\mathcal{S})$. If $\mathcal{S}$ and $\mathcal{T}$ are initialized with $\mathcal{S}_0$ and $\mathcal{T}_0$ respectively, Algorithm 2 returns a solution $\mathcal{S}_D$ satisfying

$$\phi\big((\mathcal{S}^*\cup\mathcal{S}_0)\cap\mathcal{T}_0\big) + \phi(\mathcal{S}_0) + \phi(\mathcal{T}_0) \leq 3\cdot\phi(\mathcal{S}_D),$$

and the randomized version of Algorithm 2 returns a solution $\mathcal{S}_R$ satisfying

$$\phi\big((\mathcal{S}^*\cup\mathcal{S}_0)\cap\mathcal{T}_0\big) + \phi(\mathcal{S}_0) + \phi(\mathcal{T}_0) \leq 2\cdot\mathbb{E}[\phi(\mathcal{S}_R)].$$

Proposition 2 gives rise to the approximation guarantees proved in [4].

[1] Proposition 2 is extracted through a detailed check of the proof of Theorem I.1 in [4] though it was not presented as a separate proposition therein.

Theorem 1: For the profit maximization problem, if the profit of selecting all nodes as seeds is non-negative, i.e., $\phi(\mathcal{V}) \geq 0$, the profit of the seed set $\mathcal{S}_D$ returned by Algorithm 2 satisfies

$$\phi(\mathcal{S}_D) \geq (1/3) \cdot \max_{\mathcal{S} \subseteq \mathcal{V}} \phi(\mathcal{S}),$$

and the expected profit of the seed set $\mathcal{S}_R$ returned by the randomized version of Algorithm 2 satisfies

$$\mathbb{E}[\phi(\mathcal{S}_R)] \geq (1/2) \cdot \max_{\mathcal{S} \subseteq \mathcal{V}} \phi(\mathcal{S}).$$

Remark: Buchbinder *et al.* [4] established the above approximation guarantees for any non-negative submodular function $\phi(\cdot)$ when $\mathcal{S}$ and $\mathcal{T}$ are initialized with $\emptyset$ and $\mathcal{V}$ in the double greedy algorithms. In this case, since $(\mathcal{S}^* \cup \mathcal{S}_0) \cap \mathcal{T}_0 = (\mathcal{S}^* \cup \emptyset) \cap \mathcal{V} = \mathcal{S}^* \cap \mathcal{V} = \mathcal{S}^*$, by Proposition 2, it follows that $\phi(\mathcal{S}^*) + \phi(\emptyset) + \phi(\mathcal{V}) \leq 3 \cdot \phi(\mathcal{S}_D)$ and $\phi(\mathcal{S}^*) + \phi(\emptyset) + \phi(\mathcal{V}) \leq 2 \cdot \mathbb{E}[\phi(\mathcal{S}_R)]$. Then, the approximation guarantees of Theorem 1 are derived from the non-negativity of function $\phi(\cdot)$. It is easy to see that, in fact, only a much looser condition $\phi(\emptyset) + \phi(\mathcal{V}) \geq 0$ is required to provide the approximation guarantees. In our profit maximization problem, since $\phi(\emptyset) = 0$, we just need the condition $\phi(\mathcal{V}) \geq 0$.

C. Warm-Start by Iterative Pruning

Theorem 1 provides strong theoretical guarantees for the approximabilities of Algorithms 2. However, the condition of $\phi(\mathcal{V}) \geq 0$ may not be realistic in our profit maximization problem. $\phi(\mathcal{V}) \geq 0$ means that selecting all the nodes (users) as seeds is still profitable, which is unlikely to be true for viral marketing, particularly in large-scale social networks. Apparently, providing every individual with incentives to advertise a new product defeats the purpose of viral marketing. On the other hand, when $\phi(\mathcal{V}) < 0$, we cannot have any bounded approximation guarantees by directly applying Algorithm 2. For example, consider the social network in Figure 2 with $n+1$ nodes ($n \geq 3$) and n edges. Let propagation probabilities $p_{u,v_i} = 1$, and let seed selection costs $c(u) = \frac{n}{2} + 2$ and $c(v_i) = 2$ for each $1 \leq i \leq n$. Then, $\phi(\mathcal{V}) = (n+1) - (\frac{n}{2} + 2 + 2n) = -\frac{3n}{2} - 1 < 0$. Assume that Algorithm 2 initializes $\mathcal{S} = \emptyset$ and $\mathcal{T} = \mathcal{V}$, and it iterates through the nodes in the order of $u, v_1, v_2, \cdots, v_n$. In the first iteration, we have $\phi(u|\mathcal{S}) = \phi(\{u\}) = \sigma(\{u\}) - c(u) = n+1-(\frac{n}{2}+2) = \frac{n}{2}-1$. Meanwhile, $-\phi(u|\mathcal{T} \setminus \{u\}) = -\big(\sigma(\mathcal{V}) - \sigma(\mathcal{V} \setminus \{u\}) - c(u)\big) = -\big((n+1) - n - (\frac{n}{2}+2)\big) = \frac{n}{2}+1$. Thus, u quits from $\mathcal{T}$ so that $\mathcal{S} = \emptyset$ and $\mathcal{T} = \mathcal{V} \setminus \{u\}$. In the second iteration, $\phi(v_1|\mathcal{S}) = \sigma(\{v_1\}) - \sigma(\emptyset) - c(v_1) = 1 - 2 = -1$ and $-\phi(v_1|\mathcal{T} \setminus \{v_1\}) = -\big(\sigma(\mathcal{V} \setminus \{u\}) - \sigma(\mathcal{V} \setminus \{u, v_1\}) - c(v_1)\big) = -\big(n - (n-1) - 2\big) = 1$. By Algorithm 2, v_1 quits from $\mathcal{T}$ as well. Similarly, in each subsequent iteration, v_i quits from $\mathcal{T}$. Thus, Algorithm 2 finally returns $\mathcal{S} = \emptyset$ with profit $\phi(\mathcal{S}) = 0$. However, we already know that $\phi(\{u\}) = \frac{n}{2} - 1 > 0$. When $n \to \infty$, we have $\phi(\{u\}) \to \infty$. Therefore, the deterministic double greedy algorithm can perform arbitrarily worse than the optimal solution and does not have any bounded approximation factor when $\phi(\mathcal{V}) < 0$.

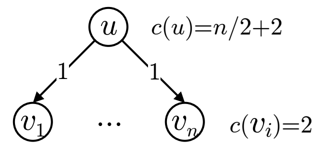


Figure 2. Double greedy algorithms fail to achieve any bounded approximation factor when $\phi(\mathcal{V}) < 0$.

To address this problem, we extend the result of Theorem 1 to maintain the same approximation guarantees with a much weaker condition. We start by proposing an approach to reduce the search space for maximizing the profit function from the power set of $\mathcal{V}$ to a smaller lattice. Given the profit function $\phi(\cdot)$, we define two node sets $\mathcal{A}_1 = \{v : \phi(v|\mathcal{V} \setminus \{v\}) > 0)\}$ and $\mathcal{B}_1 = \{v : \phi(v|\emptyset) \geq 0)\}$. Due to the submodularity of $\phi(\cdot)$, we have $\mathcal{A}_1 \subseteq \mathcal{B}_1$ and this allows us to define a lattice $\mathcal{L}_1 = [\mathcal{A}_1, \mathcal{B}_1]$ that contains all the sets $\mathcal{S}$ satisfying $\mathcal{A}_1 \subseteq \mathcal{S} \subseteq \mathcal{B}_1$. Obviously, $\mathcal{L}_1$ is a sublattice of $[\emptyset, \mathcal{V}]$.

Proposition 3: The lattice $\mathcal{L}_1 = [\mathcal{A}_1, \mathcal{B}_1]$ retains all global maximizers $\mathcal{S}^*$ for the profit function $\phi(\cdot)$, i.e., $\mathcal{A}_1 \subseteq \mathcal{S}^* \subseteq \mathcal{B}_1$ for all $\mathcal{S}^*$ where $\phi(\mathcal{S}^*) = \max_{\mathcal{S} \subseteq \mathcal{V}} \phi(\mathcal{S})$.

Proof: If $\phi(v|\mathcal{V} \setminus \{v\}) > 0$, for any seed set $\mathcal{S} \subseteq \mathcal{V} \setminus \{v\}$, it follows from the submodularity of $\phi(\cdot)$ that $\phi(v|\mathcal{S}) \geq \phi(v|\mathcal{V} \setminus \{v\}) > 0$. Thus, $\mathcal{S} \cup \{v\}$ always generates higher profit than $\mathcal{S}$, so v must be selected as a seed in every optimal solution, which indicates $\mathcal{A}_1 \subseteq \mathcal{S}^*$. By similar arguments, if $\phi(v|\emptyset) < 0$, then v cannot be selected as a seed in any optimal solution, which implies $\mathcal{S}^* \subseteq \mathcal{B}_1$. ■

Proposition 3 helps to find the nodes that must be selected as seeds and eliminate the nodes that are impossible to be chosen as seeds in an optimal solution. Thus, the lattice $\mathcal{L}_1$ is useful for warm-starts. We can prune $\mathcal{L}_1 = [\mathcal{A}_1, \mathcal{B}_1]$ even further using an iterative strategy. Specifically, since the nodes in $\mathcal{A}_1$ must be included in any global maximizer, we can shrink $\mathcal{B}_1$ to $\mathcal{B}_2 = \{v : \phi(v|\mathcal{A}_1) \geq 0\}$. Similarly, since the nodes in $\mathcal{V} \setminus \mathcal{B}_1$ cannot be included in any global maximizer, we can expand $\mathcal{A}_1$ to $\mathcal{A}_2 = \{v : \phi(v|\mathcal{B}_1 \setminus \{v\}) > 0\}$. This yields a smaller lattice $\mathcal{L}_2 = [\mathcal{A}_2, \mathcal{B}_2]$ than $\mathcal{L}_1$. These operations can be repeated alternately until $\mathcal{A}$ and $\mathcal{B}$ cannot be further broadened and narrowed respectively. Algorithm 3 presents the pseudo code of the iterative pruning process. Let $\mathcal{A}^*$ and $\mathcal{B}^*$ denote the node sets finally returned by Algorithm 3. Theorem 2 proves that the lattice $\mathcal{L}^* = [\mathcal{A}^*, \mathcal{B}^*]$ retains all global maximizers. For notational convenience, we define

Algorithm 3: $\textit{IterativePrune}(\mathcal{G}, \phi)$

```
start with t = 0, A_0 ← ∅, B_0 ← V;
repeat
    A_{t+1} ← {v : φ(v|B_t \ {v}) > 0)};
    B_{t+1} ← {v : φ(v|A_t) ≥ 0)};
    t ← t + 1;
until converged, i.e., A_t = A_{t−1} and B_t = B_{t−1};
return A_t and B_t;
```

$\mathcal{A}_0 = \emptyset$ and $\mathcal{B}_0 = \mathcal{V}$. We also make use of the following proposition about submodular functions.

Proposition 4 ([21]): For any submodular function $\phi(\cdot)$ on the power set of $\mathcal{V}$ and any two subsets $\mathcal{X}, \mathcal{Y} \subseteq \mathcal{V}$, it holds that

$$\phi(\mathcal{Y}) \leq \phi(\mathcal{X}) - \sum_{v \in \mathcal{X} \setminus \mathcal{Y}} \phi(v|\mathcal{X} \cup \mathcal{Y} \setminus \{v\}) + \sum_{v \in \mathcal{Y} \setminus \mathcal{X}} \phi(v|\mathcal{X}), \quad (1)$$

$$\phi(\mathcal{Y}) \leq \phi(\mathcal{X}) - \sum_{v \in \mathcal{X} \setminus \mathcal{Y}} \phi(v|\mathcal{X} \setminus \{v\}) + \sum_{v \in \mathcal{Y} \setminus \mathcal{X}} \phi(v|\mathcal{X} \cap \mathcal{Y}). \quad (2)$$

Theorem 2: For any global maximizer $\mathcal{S}^*$, it holds that $\mathcal{A}_t \subseteq \mathcal{A}_{t+1} \subseteq \mathcal{A}^* \subseteq \mathcal{S}^* \subseteq \mathcal{B}^* \subseteq \mathcal{B}_{t+1} \subseteq \mathcal{B}_t$ for any $t \geq 0$. Moreover, both $\phi(\mathcal{A}_t)$ and $\phi(\mathcal{B}_t)$ are non-decreasing with t.

Proof: We first show that after each iteration, the newly generated lattice is a sublattice of that in the previous iteration, i.e., $\mathcal{A}_t \subseteq \mathcal{A}_{t+1} \subseteq \mathcal{B}_{t+1} \subseteq \mathcal{B}_t$. We prove it by induction. It holds obviously that $\mathcal{A}_0 = \emptyset \subseteq \mathcal{A}_1 \subseteq \mathcal{B}_1 \subseteq \mathcal{V} = \mathcal{B}_0$ according to Proposition 3. Suppose that $\mathcal{A}_{t-1} \subseteq \mathcal{A}_t \subseteq \mathcal{B}_t \subseteq \mathcal{B}_{t-1}$ for some $t \geq 1$. For every node $v \in \mathcal{A}_t$, we know $\phi(v|\mathcal{B}_{t-1} \setminus \{v\}) > 0$. Due to the submodularity, we have $\phi(v|\mathcal{B}_t \setminus \{v\}) \geq \phi(v|\mathcal{B}_{t-1} \setminus \{v\}) > 0$. As a result, $\mathcal{A}_t \subseteq \mathcal{A}_{t+1}$. Similarly, for every node $v \in \mathcal{B}_{t+1}$, we have $\phi(v|\mathcal{A}_t) \geq 0$. Due to the submodularity, $\phi(v|\mathcal{A}_{t-1}) \geq \phi(v|\mathcal{A}_t) \geq 0$, which indicates that $\mathcal{B}_{t+1} \subseteq \mathcal{B}_t$. Furthermore, for all nodes $v \in \mathcal{A}_{t+1} \cap \mathcal{A}_t$, we have $\phi(v|\mathcal{A}_t) = 0$, which implies that $(\mathcal{A}_{t+1} \cap \mathcal{A}_t) \subseteq \mathcal{B}_{t+1}$ via line 4 in Algorithm 3. For all nodes $v \in \mathcal{A}_{t+1} \setminus \mathcal{A}_t$, we have $\phi(v|\mathcal{B}_t \setminus \{v\}) > 0$. Since $v \notin \mathcal{A}_t$ and $\mathcal{A}_t \subseteq \mathcal{B}_t$, we also have $\mathcal{A}_t \subseteq \mathcal{B}_t \setminus \{v\}$. Thus, $\phi(v|\mathcal{A}_t) \geq \phi(v|\mathcal{B}_t \setminus \{v\}) > 0$, which implies that $(\mathcal{A}_{t+1} \setminus \mathcal{A}_t) \subseteq \mathcal{B}_{t+1}$. Consequently, it holds that $\mathcal{A}_{t+1} = (\mathcal{A}_{t+1} \cap \mathcal{A}_t) \cup (\mathcal{A}_{t+1} \setminus \mathcal{A}_t) \subseteq \mathcal{B}_{t+1}$. Therefore, $\mathcal{A}_t \subseteq \mathcal{A}_{t+1} \subseteq \mathcal{B}_{t+1} \subseteq \mathcal{B}_t$ holds for any $t \geq 0$.

Now, let us turn to exploring the relationship of $\mathcal{S}^*$ to $\mathcal{A}^*$ and $\mathcal{B}^*$. Obviously, $\mathcal{A}_0 = \emptyset \subseteq \mathcal{S}^* \subseteq \mathcal{V} = \mathcal{B}_0$ holds. As proved in Proposition 3, $\mathcal{A}_1 \subseteq \mathcal{S}^* \subseteq \mathcal{B}_1$ also holds. Suppose that $\mathcal{A}_t \subseteq \mathcal{S}^* \subseteq \mathcal{B}_t$ for some $t \geq 0$. Then, any node v satisfying $\phi(v|\mathcal{B}_t \setminus \{v\}) > 0$ must be in $\mathcal{S}^*$. Otherwise, if $v \notin \mathcal{S}^*$, we have $\phi(v|\mathcal{S}^*) = \phi(v|\mathcal{S}^* \setminus \{v\}) \geq \phi(v|\mathcal{B}_t \setminus \{v\}) > 0$ by the submodularity, which indicates $\phi(\mathcal{S}^* \cup \{v\}) > \phi(\mathcal{S}^*)$, contradicting the optimality of $\mathcal{S}^*$. The set of such nodes v satisfying $\phi(v|\mathcal{B}_t \setminus \{v\}) > 0$ is exactly $\mathcal{A}_{t+1}$, and hence $\mathcal{A}_{t+1} \subseteq \mathcal{S}^*$. Similarly, any node $v \in \mathcal{S}^*$ must be in $\mathcal{B}_{t+1}$, which means $\mathcal{S}^* \subseteq \mathcal{B}_{t+1}$. Otherwise, if $v \notin \mathcal{B}_{t+1}$, it holds that $\phi(v|\mathcal{A}_t) < 0$ by the definition, and hence $v \notin \mathcal{A}_t$. As a result, $\phi(v|\mathcal{S}^* \setminus \{v\}) \leq \phi(v|\mathcal{A}_t \setminus \{v\}) = \phi(v|\mathcal{A}_t) < 0$ by the submodularity, which implies $\phi(\mathcal{S}^*) < \phi(\mathcal{S}^* \setminus \{v\})$, contradicting the optimality of $\mathcal{S}^*$. By induction, we have $\mathcal{A}_t \subseteq \mathcal{S}^* \subseteq \mathcal{B}_t$ for any $t \geq 0$ and thus, $\mathcal{A}^* \subseteq \mathcal{S}^* \subseteq \mathcal{B}^*$.

Finally, we show that $\phi(\mathcal{A}_t) \leq \phi(\mathcal{A}_{t+1})$ and $\phi(\mathcal{B}_t) \leq \phi(\mathcal{B}_{t+1})$ for any $t \geq 0$. In fact, for any node $v \in \mathcal{A}_{t+1} \setminus \mathcal{A}_t$, it holds that $\phi(v|\mathcal{A}_{t+1} \setminus \{v\}) \geq \phi(v|\mathcal{B}_{t+1} \setminus \{v\}) \geq \phi(v|\mathcal{B}_t \setminus \{v\}) > 0$, where the first two inequalities are due to the submodularity (since $\mathcal{A}_{t+1} \subseteq \mathcal{B}_{t+1} \subseteq \mathcal{B}_t$) and the third inequality is by the definition of $\mathcal{A}_{t+1}$. Therefore, $\phi(\mathcal{A}_{t+1}) \geq \phi(\mathcal{A}_t) + \sum_{v \in \mathcal{A}_{t+1} \setminus \mathcal{A}_t} \phi(v|\mathcal{A}_{t+1} \setminus \{v\}) \geq \phi(\mathcal{A}_t)$, where the first inequality is due to (1) of Proposition 4 and the fact $\mathcal{A}_t \subseteq \mathcal{A}_{t+1}$. Similarly, for any node $v \in \mathcal{B}_t \setminus \mathcal{B}_{t+1}$, we have $\phi(v|\mathcal{B}_{t+1}) \leq \phi(v|\mathcal{A}_{t+1}) \leq \phi(v|\mathcal{A}_t) < 0$, where the first two inequalities are due to the submodularity (since $\mathcal{A}_t \subseteq \mathcal{A}_{t+1} \subseteq \mathcal{B}_{t+1}$) and the third inequality is by the definition of $\mathcal{B}_{t+1}$ as $v \notin \mathcal{B}_{t+1}$. Hence, $\phi(\mathcal{B}_t) \leq \phi(\mathcal{B}_{t+1}) + \sum_{v \in \mathcal{B}_t \setminus \mathcal{B}_{t+1}} \phi(v|\mathcal{B}_{t+1}) \leq \phi(\mathcal{B}_{t+1})$, where the first inequality is due to (2) of Proposition 4 and the fact $\mathcal{B}_{t+1} \subseteq \mathcal{B}_t$. ■

Now, instead of starting with $\mathcal{S} = \emptyset$ and $\mathcal{T} = \mathcal{V}$ (the entire node set) in the double greedy algorithms, we can initialize $\mathcal{S}$ with $\mathcal{A}^*$ and $\mathcal{T}$ with $\mathcal{B}^*$ and only check the nodes in $\mathcal{B}^* \setminus \mathcal{A}^*$ to decide whether or not to include them in $\mathcal{S}$ and $\mathcal{T}$. The following corollary establishes the approximation guarantees for the modified algorithms.

Corollary 1: Suppose that $\mathcal{S}$ and $\mathcal{T}$ are initialized with $\mathcal{A}^*$ and $\mathcal{B}^*$ such that $\phi(\mathcal{A}^*) + \phi(\mathcal{B}^*) \geq 0$, the profit of the seed set $\hat{\mathcal{S}}_D$ returned by Algorithm 2 satisfies

$$\phi(\hat{\mathcal{S}}_D) \geq (1/3) \cdot \max_{\mathcal{S} \subseteq \mathcal{V}} \phi(\mathcal{S}),$$

and the expected profit of the seed set $\hat{\mathcal{S}}_R$ returned by the randomized version of Algorithm 2 satisfies

$$\mathbb{E}[\phi(\hat{\mathcal{S}}_R)] \geq (1/2) \cdot \max_{\mathcal{S} \subseteq \mathcal{V}} \phi(\mathcal{S}).$$

Proof: By Theorem 2, $\mathcal{A}^* \subseteq \mathcal{S}^* \subseteq \mathcal{B}^*$. Thus, $(\mathcal{S}^* \cup \mathcal{A}^*) \cap \mathcal{B}^* = \mathcal{S}^* \cap \mathcal{B}^* = \mathcal{S}^*$. As a result, by Proposition 2, it holds that $\phi(\mathcal{S}^*) + \phi(\mathcal{A}^*) + \phi(\mathcal{B}^*) \leq 3 \cdot \phi(\hat{\mathcal{S}}_D)$ when $\mathcal{S}$ and $\mathcal{T}$ are initialized with $\mathcal{A}^*$ and $\mathcal{B}^*$. Hence, if $\phi(\mathcal{A}^*) + \phi(\mathcal{B}^*) \geq 0$, we obtain $\phi(\hat{\mathcal{S}}_D) \geq (1/3) \cdot \phi(\mathcal{S}^*) = (1/3) \cdot \max_{\mathcal{S} \subseteq \mathcal{V}} \phi(\mathcal{S})$. The proof of $\mathbb{E}[\phi(\hat{\mathcal{S}}_R)] \geq (1/2) \cdot \max_{\mathcal{S} \subseteq \mathcal{V}} \phi(\mathcal{S})$ is similar. ■

Corollary 1 shows that using an iterative pruning approach prior to applying the double greedy algorithms allows us to maintain the same approximation guarantees with the condition $\phi(\mathcal{A}^*) + \phi(\mathcal{B}^*) \geq 0$. This condition is much weaker than the original condition $\phi(\mathcal{V}) \geq 0$ of Theorem 1 since by Theorem 2, $\phi(\mathcal{V}) = \phi(\emptyset) + \phi(\mathcal{V}) = \phi(\mathcal{A}_0) + \phi(\mathcal{B}_0) \leq \phi(\mathcal{A}_1) + \phi(\mathcal{B}_1) \leq \cdots \leq \phi(\mathcal{A}^*) + \phi(\mathcal{B}^*)$. Thus, Corollary 1 significantly expands the applicability of the theoretical guarantees.

For the example shown in Figure 2, $\phi(u|\mathcal{V} \setminus \{u\}) = 1 - (\frac{n}{2} + 2) < 0$ and $\phi(v_i|\mathcal{V} \setminus \{v_i\}) = 0 - 2 < 0$ for each $1 \leq i \leq n$. Thus, by Algorithm 3, $\mathcal{A}_1 = \emptyset$. Furthermore, $\phi(u|\emptyset) = n + 1 - (\frac{n}{2} + 1) = \frac{n}{2} - 1 > 0$ and $\phi(v_i|\emptyset) = 1 - 2 = -1 < 0$ for each $1 \leq i \leq n$, which implies $\mathcal{B}_1 = \{u\}$. In the second iteration, since $\phi(u|\mathcal{B}_1 \setminus \{u\}) = \phi(u|\emptyset) = \frac{n}{2} - 1 > 0$, we have $\mathcal{A}_2 = \{u\}$. $\mathcal{B}_2$ remains the same as $\mathcal{B}_1 = \{u\}$. Thus, Algorithm 3 returns $\mathcal{A}^* = \mathcal{B}^* = \{u\}$. As a result, $\phi(\mathcal{A}^*) + \phi(\mathcal{B}^*) = 2 \cdot \phi(\{u\}) = n - 2 \geq 0$ satisfying the condition given in Corollary 1. In fact, since $\mathcal{A}^* = \mathcal{B}^*$, it must be an optimal solution and the double greedy algorithms return exactly the optimal solution.

Finally, we remark that the sets $\mathcal{A}^*$ and $\mathcal{B}^*$ produced by iterative pruning can also be used as warm-starts for the simple greedy algorithm in Section IV-A so that only the nodes in $\mathcal{B}^* \setminus \mathcal{A}^*$ need to be further examined for seed selection.

D. Online Upper Bounds

Besides the approximation guarantees provided by Corollary 1 for the double greedy algorithms, we can also derive online upper bounds on the optimal solution to the profit maximization problem without any condition requirement. Following the proof of Corollary 1, we directly have two upper bounds below,

$$\mu_1 \triangleq 3 \cdot \phi(\hat{\mathcal{S}}_D) - [\phi(\mathcal{A}^*) + \phi(\mathcal{B}^*)] \geq \phi(\mathcal{S}^*), \quad (3)$$

$$\mu_2 \triangleq 2 \cdot \mathbb{E}[\phi(\hat{\mathcal{S}}_R)] - [\phi(\mathcal{A}^*) + \phi(\mathcal{B}^*)] \geq \phi(\mathcal{S}^*). \quad (4)$$

Regardless of whether $\phi(\mathcal{A}^*) + \phi(\mathcal{B}^*)$ is negative or non-negative, the above upper bounds always hold. Moreover, when $\phi(\mathcal{A}^*) + \phi(\mathcal{B}^*) > 0$, these bounds are tighter than the constant approximation factors of $(1/3)$ and $(1/2)$ for the double greedy algorithms.

Upper bounds can also be derived solely based on the submodularity of the function. Consider the two relations in Proposition 4. Restricting the sets $\mathcal{X}$ and $\mathcal{Y}$ to the sublattice $\mathcal{L}^* = [\mathcal{A}^*, \mathcal{B}^*]$ (i.e., $\mathcal{A}^* \subseteq \mathcal{X}, \mathcal{Y} \subseteq \mathcal{B}^*$), we define two modular upper bounds on $\phi(\mathcal{Y})$ as follows:

$$m_{\mathcal{X}}(\mathcal{Y}) \triangleq \phi(\mathcal{X}) - \sum_{v \in \mathcal{X} \setminus \mathcal{Y}} \phi(v|\mathcal{B}^* \setminus \{v\}) + \sum_{v \in \mathcal{Y} \setminus \mathcal{X}} \phi(v|\mathcal{X}), \quad (5)$$

$$\bar{m}_{\mathcal{X}}(\mathcal{Y}) \triangleq \phi(\mathcal{X}) - \sum_{v \in \mathcal{X} \setminus \mathcal{Y}} \phi(v|\mathcal{X} \setminus \{v\}) + \sum_{v \in \mathcal{Y} \setminus \mathcal{X}} \phi(v|\mathcal{A}^*). \quad (6)$$

Proposition 5: For any two sets $\mathcal{X}$ and $\mathcal{Y}$ where $\mathcal{A}^* \subseteq \mathcal{X}, \mathcal{Y} \subseteq \mathcal{B}^*$, the above defined $m_{\mathcal{X}}(\mathcal{Y})$ and $\bar{m}_{\mathcal{X}}(\mathcal{Y})$ satisfy

$$m_{\mathcal{X}}(\mathcal{Y}) \geq \phi(\mathcal{Y}) \quad \text{and} \quad \bar{m}_{\mathcal{X}}(\mathcal{Y}) \geq \phi(\mathcal{Y}).$$

Proof: For each node $v \in \mathcal{X} \setminus \mathcal{Y}$, since $\mathcal{X} \cup \mathcal{Y} \setminus \{v\} \subseteq \mathcal{B}^* \setminus \{v\}$, it holds that $\phi(v|\mathcal{B}^* \setminus \{v\}) \leq \phi(v|\mathcal{X} \cup \mathcal{Y} \setminus \{v\})$ due to the submodularity. By (1) and (5), we obtain $m_{\mathcal{X}}(\mathcal{Y}) \geq \phi(\mathcal{Y})$. Similarly, for each node $v \in \mathcal{Y} \setminus \mathcal{X}$, since $\mathcal{A}^* \subseteq \mathcal{X} \cap \mathcal{Y}$, it holds that $\phi(v|\mathcal{A}^*) \geq \phi(v|\mathcal{X} \cap \mathcal{Y})$. By (2) and (6), we obtain $\bar{m}_{\mathcal{X}}(\mathcal{Y}) \geq \phi(\mathcal{Y})$. ■

Based on Proposition 5, we can find two series of upper bounds on the maximum value of $\phi(\cdot)$ as follows. For any set $\mathcal{X}$ where $\mathcal{A}^* \subseteq \mathcal{X} \subseteq \mathcal{B}^*$, we define

$$\mu(\mathcal{X}) \triangleq \max_{\mathcal{A}^* \subseteq \mathcal{Y} \subseteq \mathcal{B}^*} m_{\mathcal{X}}(\mathcal{Y}) \quad \text{and} \quad \bar{\mu}(\mathcal{X}) \triangleq \max_{\mathcal{A}^* \subseteq \mathcal{Y} \subseteq \mathcal{B}^*} \bar{m}_{\mathcal{X}}(\mathcal{Y}).$$

Theorem 3: For any set $\mathcal{X}$ where $\mathcal{A}^* \subseteq \mathcal{X} \subseteq \mathcal{B}^*$,

$$\mu(\mathcal{X}) \geq \phi(\mathcal{S}^*) \quad \text{and} \quad \bar{\mu}(\mathcal{X}) \geq \phi(\mathcal{S}^*).$$

Proof: The proof is straightforward since $\mu(\mathcal{X}) \geq m_{\mathcal{X}}(\mathcal{S}^*) \geq \phi(\mathcal{S}^*)$, where the first inequality is by the definition of $\mu(\mathcal{X})$ and the second inequality is due to Proposition 5. Similarly, we have $\bar{\mu}(\mathcal{X}) \geq \bar{m}_{\mathcal{X}}(\mathcal{S}^*) \geq \phi(\mathcal{S}^*)$. ■

The upper bounds established above can be computed very fast since $m_{\mathcal{X}}(\mathcal{Y})$ and $\bar{m}_{\mathcal{X}}(\mathcal{Y})$ are modular functions with respect to $\mathcal{Y}$. It is much easier to find the maximum value for a modular function than that for a submodular function. We can obtain upper bounds by arbitrarily choosing the set $\mathcal{X}$ in $\mu(\mathcal{X})$ and $\bar{\mu}(\mathcal{X})$. In particular, we consider setting $\mathcal{X}$ to the seed sets returned by the simple greedy and double greedy algorithms. It can be proved that the upper bounds derived from such sets $\mathcal{X}$ are guaranteed to be tighter than those derived by setting $\mathcal{X} = \mathcal{A}^*$ or $\mathcal{X} = \mathcal{B}^*$ (detailed proofs are omitted due to space limitations). Given the seed set $\mathcal{S}^g$ returned by a greedy algorithm, we have the following two upper bounds.

$$\mu_3 \triangleq \mu(\mathcal{S}^g), \quad (7)$$

$$\mu_4 \triangleq \bar{\mu}(\mathcal{S}^g). \quad (8)$$

E. Extension to Other Influence Propagation Models

Our analysis and algorithms are general frameworks that can be adapted to any influence propagation models which are submodular. In fact, there are many submodular influence propagation models.

The *triggering model* [16] generalizes the IC model and another widely used LT model [26], [27]. It randomly assigns each node v with a subset $\mathcal{T}_v$ of its inverse neighbors according to a *triggering distribution*. The diffusion process works as follows. All the nodes in the seed set $\mathcal{S}$ are initially activated. Any other node v would become activated if any inverse neighbor in its triggering set $\mathcal{T}_v$ is activated. This process repeats until no more node can be activated.

Continuous-time models [9], [23] assume that the influence propagates through the OSN within a time window. In these models, each edge $e \in \mathcal{E}$ is associated with a transmission time (or length) distribution $t(e)$. The nodes can only be activated within the time window since the propagation starts from seed nodes.

Topic-aware models [3], [5] consider each edge (u, v) to be associated with a propagation probability $p^z_{u,v}$ on each topic z. A node u can influence each of its neighbors v via a mixed probability dependent on the topics involved in queries.

Our solutions and analysis can be applied to all the above models.

F. Time Complexity

Evaluating the profit metric involves estimating the influence spread given a seed set. Any existing influence estimation methods, such as Monte-Carlo simulation [16], [17], [22], [24] and Reverse Influence Sampling (RIS) [26], [27], can be used. The effectiveness and efficiency of influence estimation are beyond the scope of this paper. Suppose the time complexity for computing the marginal profit gain of adding u into $\mathcal{S}$ or removing u from $\mathcal{T}$ is $O(M)$. The simple greedy algorithm (Algorithm 1) takes at most $O\big((|\mathcal{V}| - |\mathcal{S}|)M\big)$ time to select one seed. Thus, the total time complexity of the simple greedy algorithm is $O\big((|\mathcal{V}| + |\mathcal{V}| - 1 + \cdots + 1)M\big) = O(|\mathcal{V}|^2 M)$. The double greedy algorithm (Algorithm 2) takes $O(2M)$ time for checking each node to decide whether to select it as a seed. Thus, the total time complexity of the double greedy algorithm is $O(|\mathcal{V}| \cdot 2M) = O(|\mathcal{V}|M)$. For the iterative pruning process (Algorithm 3), the size of the node set $\mathcal{B}_t \setminus \mathcal{A}_t$ to check reduces by at least 1 in each iteration. Therefore, it has a time complexity of $O(|\mathcal{V}|^2 M)$.

V. Evaluation

A. Experimental Setup

Datasets. We use several real OSN datasets in our experiments [1]. Table I shows the statistics of these datasets.

Table I
STATISTICS OF OSN DATASETS.

Dataset	#Nodes ($|\mathcal{V}|$)	#Edges ($|\mathcal{E}|$)	Avg. degree
Facebook	4K	176K	43.7
Wiki-Vote	7K	10K	14.6
Google+	108K	14M	127.1
LiveJournal	5M	69M	14.2

Algorithms. The performance comparison includes the following algorithms.

- *Random* (Rand): It randomly selects k nodes. We run the algorithm 10 times and take their average as the expected profit. To explore how k would affect the result, we iterate through $k = \frac{|\mathcal{V}|}{2^i}$ for $i = 0, 1, \ldots, 10$ and choose the one with the largest expected profit.
- *High Degree* (HD): It selects k nodes with the highest degrees. Similar to the random algorithm, we also iterate through different k values and choose the one producing the largest profit among $k = \frac{|\mathcal{V}|}{2^i}$ for $i = 0, 1, \ldots, 10$.
- *IMM*: IMM is a state-of-the-art sampling-based method that can provide theoretical guarantees for finding the top-k influential nodes for influence maximization. We set its algorithm parameters $\epsilon = 0.5$ and $l = 1$ according to the default setting in [26]. We also choose the k value producing the largest profit among $k = \frac{|\mathcal{V}|}{2^i}$ for $i = 0, 1, \ldots, 10$.
- *Simple Greedy* (SG): We adopt the *Reverse Influence Sampling* (RIS) method used in IMM for influence estimation in our proposed algorithms. The number of Reverse Reachable (RR) sets is set to the maximal number of RR sets generated in the IMM method among the above 11 cases of different k values. We also adopt the CELF technique [17] in the implementation of the simple greedy algorithm to enhance its efficiency.
- *Simple Greedy with Iterative Pruning* (SGIP): It runs simple greedy after the iterative pruning procedure described in Section IV-C.
- *Double Greedy* (DG): We generate the same number of RR sets as that for SG. Since the deterministic and randomized double greedy algorithms perform quite similarly in terms of the profit generated and the running time taken, we shall report the results of only the deterministic algorithm in order to make the figures easier to read.
- *Double Greedy with Iterative Pruning* (DGIP): It runs double greedy after the iterative pruning procedure.

Parameter Settings. For the IC model, we set the propagation probability $p_{u,v}$ of each edge (u, v) to the reciprocal of v's in-degree, i.e., $p_{u,v} = 1/|\mathcal{I}_v|$, as widely adopted by other studies [6], [15], [26], [27]. The average seed selection cost of all nodes is set equal to a scale factor λ, i.e., $\sum_{v \in \mathcal{V}} c(v) = \lambda \cdot |\mathcal{V}|$. The larger the factor λ, the higher the cost of seed selection relative to the benefit of influence spread. The default value of λ is set to 10.[2] We consider two cost distributions. In the uniform setting, all the nodes have the same costs for seed selection. In the degree-proportional setting, the cost of each node is set proportional to its out-degree to emulate that popular users need more incentives to participate. To evaluate the profits of the seed sets returned by different algorithms, we estimate the influence spread of each seed set by taking the average measurement of $10,000$ Monte-Carlo simulations.

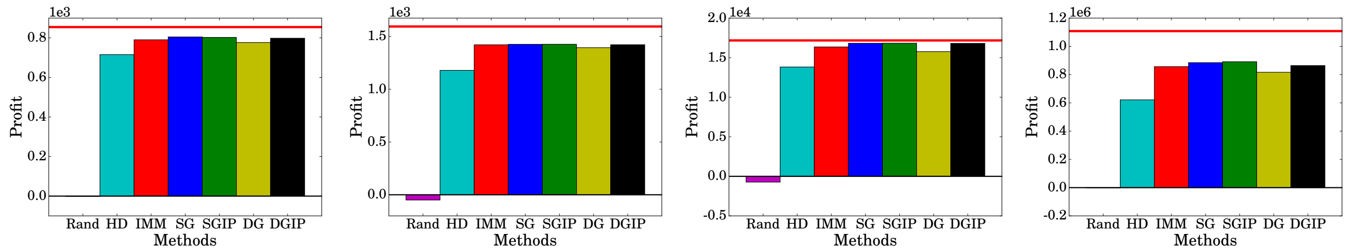
[2]We have tested other values of λ and observed similar performance trends. Only the results for $\lambda = 10$ are presented due to space limitations.

B. Results

Figures 3 and 4 show the profits produced by different algorithms under different cost distributions. The red horizontal line is the tightest online upper bound among those described in Section IV-D. We shall further compare different online upper bounds later.

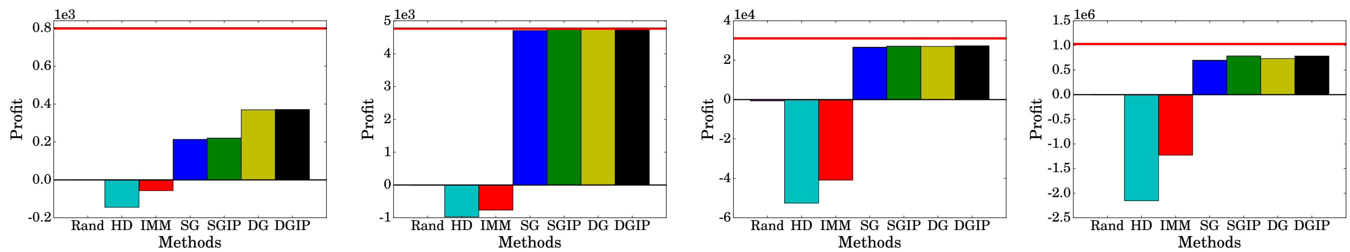
Comparing the seed selection algorithms, our greedy algorithms are more effective in optimizing the profit than the three baseline algorithms (Rand, HD and IMM) on the datasets tested. We also observe that the greedy algorithms usually perform quite close to the optimal solution, which confirms the effectiveness of our proposed techniques. Under the uniform cost distribution, as seen from Figure 3, the greedy algorithms and the IMM algorithm produce much higher profits than the high degree algorithm, whereas the random algorithm generates little profit and is difficult to benefit from viral marketing. In fact, when all nodes have the same costs for seed selection, our simple greedy algorithm degenerates to the IMM algorithm that iterates through different seed set sizes. Under the degree-proportional cost distribution, as seen from Figure 4, the high degree and IMM algorithms perform even worse than the random algorithm with very negative profits produced. This is because under such a setting, the nodes with high influence have large costs. Therefore, the expense would be larger than the revenue obtained when selecting too many such nodes as seeds. We can also see that the double greedy algorithms perform considerably better than the simple greedy algorithms on the Facebook dataset, which confirms that the simple greedy algorithms do not have any guarantees as discussed in Section IV-A. Moreover, the results of Figures 3 and 4 also show that the iterative pruning technique can further improve the greedy algorithms (by up to 13%).

Table II shows the running times of different algorithms. The algorithms are all implemented in C++ and the experiments are carried out on a machine with an Intel Xeon E5-1650 3.2GHz CPU and 16GB memory. As the running times for the random and high degree algorithms are very short (less than 0.01 second), we omit them in Table II. It can be seen that the IMM algorithm runs significantly slower than our greedy algorithms. This is because the IMM algorithm needs to test different seed set sizes separately to find the solution. On the other hand, the running times of different greedy algorithms are similar. This is because the major of time for running these algorithms is taken by generating the RR sets and the numbers of RR sets used by different greedy algorithms are the same. We also observe that even for the large LiveJournal



(a) Facebook (b) Wiki-Vote (c) Google+ (d) LiveJournal

Figure 3. Profits produced by different algorithms under uniform cost distribution.



(a) Facebook (b) Wiki-Vote (c) Google+ (d) LiveJournal

Figure 4. Profits produced by different algorithms under degree-proportional cost distribution.

Table II
RUNNING TIMES OF DIFFERENT ALGORITHMS (SECONDS).

Dataset	IMM	SG	SGIP	DG	DGIP
Facebook	1.28	0.21	0.24	0.22	0.24
Wiki-Vote	0.91	0.14	0.10	0.19	0.10
Google+	31.02	6.02	7.09	5.72	5.98
LiveJournal	3273.63	550.91	588.66	545.40	587.12

(a) Uniform Cost Distribution

Dataset	IMM	SG	SGIP	DG	DGIP
Facebook	2.12	0.25	0.25	0.22	0.24
Wiki-Vote	1.10	0.08	0.09	0.08	0.09
Google+	29.42	6.87	6.53	6.94	6.55
LiveJournal	2756.32	563.16	581.76	555.90	582.53

(b) Degree-Proportional Cost Distribution

Table III
IMPACT OF ITERATIVE PRUNING.

Dataset	$\lvert\mathcal{A}^*\rvert$	$\lvert\mathcal{B}^*\rvert$	$\lvert\mathcal{B}^* \setminus \mathcal{A}^*\rvert$	$\phi(\mathcal{A}^*) + \phi(\mathcal{B}^*)$
Facebook	12	158	146	622
Wiki-Vote	54	241	187	2,104
Google+	715	856	141	33,624
LiveJournal	2,719	548,855	546,136	-2,460,900

(a) Uniform Cost Distribution

	$\lvert\mathcal{A}^*\rvert$	$\lvert\mathcal{B}^*\rvert$	$\lvert\mathcal{B}^* \setminus \mathcal{A}^*\rvert$	$\phi(\mathcal{A}^*) + \phi(\mathcal{B}^*)$
Facebook	53	2,589	2,536	-8,678
Wiki-Vote	4,808	4,808	0	9,537
Google+	36,070	43,062	6,992	54,947
LiveJournal	1,136,106	1,738,332	602,226	1,372,225

(b) Degree-Proportional Cost Distribution

dataset with millions of nodes, the running times of our greedy algorithms are less than 600 seconds, which demonstrates the efficiency of our algorithms.

Table III summarizes the impact of the iterative pruning technique proposed in Section IV-C. As can be seen, pruning substantially reduces the number of nodes that need to be considered for seed selection. In addition, with a scale factor $\lambda = 10$ for the seed selection cost, the profit of selecting all the nodes is $\phi(\mathcal{V}) = |\mathcal{V}| - 10 \cdot |\mathcal{V}| < 0$. Thus, running the double greedy algorithm with the entire node set would not offer any approximation guarantee. In contrast, as shown in Table III, $\phi(\mathcal{A}^*) + \phi(\mathcal{B}^*) > 0$ holds for most of the cases tested. In these cases, the pruning technique enables strong theoretical guarantees on the seed sets constructed by double greedy algorithms according to Corollary 1. In particular, for the degree-proportional cost distribution on Wiki-Vote, we have $\mathcal{A}^* = \mathcal{B}^*$ so that no node needs to be further checked for seed selection after pruning, which implies that the pruning process directly produces the optimal seed set for profit maximization.

Table IV shows the online upper bounds derived from the seed sets returned by the double greedy algorithms without/with iterative pruning. To quantify their relative order, these bounds are normalized by the actual profit produced by the DGIP algorithm. As can be seen, for all the datasets tested, μ_1 is always the loosest upper bound among all those obtained, while $\mu_3 = \mu(\mathcal{S}^g)$ is always the tightest one no matter whether the pruning technique is used. Comparing the upper bounds derived from the DG and DGIP seed sets, the latter are considerably lower than the former. This implies that the iterative pruning technique can improve the bounds significantly. Next, we examine the upper bounds derived from the DGIP seed sets in detail (right half of Table IV). For the cases where $\phi(\mathcal{A}^*) + \phi(\mathcal{B}^*) < 0$ (see Table III), μ_1^{DGIP} is far above 3 times the profit returned by the algorithm. In these

Table IV
NORMALIZED ONLINE UPPER BOUNDS.

Dataset	μ_1^{DG}	μ_3^{DG}	μ_4^{DG}	μ_1^{DGIP}	μ_3^{DGIP}	μ_4^{DGIP}
Facebook	49.79	1.21	8.80	2.26	**1.07**	1.25
Wiki-Vote	49.00	1.63	3.94	1.62	**1.12**	1.28
Google+	64.45	1.37	3.50	1.05	**1.02**	1.03
LiveJournal	56.44	1.51	26.21	5.87	**1.28**	5.64

(a) Uniform Cost Distribution

Dataset	μ_1^{DG}	μ_3^{DG}	μ_4^{DG}	μ_1^{DGIP}	μ_3^{DGIP}	μ_4^{DGIP}
Facebook	101.99	2.31	12.76	27.05	**2.15**	11.22
Wiki-Vote	16.45	1.00	1.02	1.00	**1.00**	1.00
Google+	39.17	1.34	1.59	1.30	**1.14**	1.20
LiveJournal	63.30	2.30	7.89	1.70	**1.31**	1.31

(b) Degree-Proportional Cost Distribution

cases, μ_3^{DGIP} certifies approximation guarantees from 46% to 78% for the seed set returned by the DGIP algorithm. For other cases where $\phi(\mathcal{A}^*) + \phi(\mathcal{B}^*) \geq 0$, all the bounds are rather close to the profit obtained by the DGIP algorithm, and μ_3^{DGIP} certifies at least 76% approximation guarantee for the seed set returned by the algorithm. This observation implies that the DGIP algorithm usually performs quite close to the optimal solution and confirms the effectiveness of our proposed techniques.

VI. CONCLUSION

In this paper, we have studied a profit maximization problem for viral marketing in OSNs. The objective is to select initial seed nodes to maximize the total profit that accounts for the benefit of influence spread as well as the cost of seed selection. The non-monotone characteristic of the profit metric makes the profit maximization problem more computationally challenging than the traditional influence maximization problem. We have presented simple greedy and double greedy seed selection algorithms, and proposed several new techniques to enhance the practical performance of the algorithms and expand the applicability of their approximation guarantees. Experiments are conducted with real OSN datasets to compare different algorithms. The results show that (1) our greedy algorithms substantially outperform several baseline algorithms and (2) our iterative pruning technique can further improve the greedy algorithms and provide much tighter online upper bounds to benchmark their performance.

ACKNOWLEDGMENT

This research is supported by the National Research Foundation, Prime Minister's Office, Singapore under its IDM Futures Funding Initiative, and by Singapore Ministry of Education Academic Research Fund Tier 1 under Grant 2013-T1-002-123.

REFERENCES

[1] http://snap.stanford.edu/data/index.html.

[2] C. Aslay, W. Lu, F. Bonchi, A. Goyal, and L. V. S. Lakshmanan. Viral marketing meets social advertising: Ad allocation with minimum regret. *Proc. VLDB Endowment*, 8(7):814–825, 2015.

[3] N. Barbieri, F. Bonchi, and G. Manco. Topic-aware social influence propagation models. In *Proc. IEEE ICDM*, pages 81–90, 2012.

[4] N. Buchbinder, M. Feldman, J. S. Naor, and R. Schwartz. A tight linear time (1/2)-approximation for unconstrained submodular maximization. In *Proc. IEEE FOCS*, pages 649–658, 2012.

[5] S. Chen, J. Fan, G. Li, J. Feng, K.-l. Tan, and J. Tang. Online topic-aware influence maximization. *Proc. VLDB Endow.*, pages 666–677, 2015.

[6] W. Chen, C. Wang, and Y. Wang. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *Proc. ACM KDD*, pages 1029–1038, 2010.

[7] W. Chen, Y. Wang, and S. Yang. Efficient influence maximization in social networks. In *Proc. ACM KDD*, pages 199–208, 2009.

[8] P. Domingos and M. Richardson. Mining the network value of customers. In *Proc. ACM KDD*, pages 57–66, 2001.

[9] N. Du, L. Song, M. Gomez-Rodriguez, and H. Zha. Scalable influence estimation in continuous-time diffusion networks. In *Proc. NIPS*, pages 3147–3155, 2013.

[10] U. Feige, V. S. Mirrokni, and J. Vondrák. Maximizing non-monotone submodular functions. In *Proc. IEEE FOCS*, pages 461–471, 2007.

[11] A. Goyal, F. Bonchi, L. Lakshmanan, and S. Venkatasubramanian. On minimizing budget and time in influence propagation over social networks. *Social Network Analysis and Mining*, 3(2):179–192, 2013.

[12] J. Hartline, V. Mirrokni, and M. Sundararajan. Optimal marketing strategies over social networks. In *Proc. WWW*, pages 189–198, 2008.

[13] R. Iyer and J. Bilmes. Submodular optimization with submodular cover and submodular knapsack constraints. In *Proc. NIPS*, pages 2436–2444, 2013.

[14] R. Iyer, S. Jegelka, and J. Bilmes. Fast semidifferential-based submodular function optimization. In *Proc. ICML*, pages 855–863, 2013.

[15] K. Jung, W. Heo, and W. Chen. IRIE: Scalable and robust influence maximization in social networks. In *Proc. IEEE ICDM*, pages 918–923, 2012.

[16] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *Proc. ACM KDD*, pages 137–146, 2003.

[17] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance. Cost-effective outbreak detection in networks. In *Proc. ACM KDD*, pages 420–429, 2007.

[18] Y. Li, B. Q. Zhao, and J. C. S. Lui. On modeling product advertisement in large-scale online social networks. *IEEE/ACM Trans. Networking*, 20(5):1412–1425, 2012.

[19] C. Long and R. C.-W. Wong. Minimizing seed set for viral marketing. In *Proc. IEEE ICDM*, pages 427–436, 2011.

[20] W. Lu and L. V. Lakshmanan. Profit maximization over social networks. In *Proc. IEEE ICDM*, pages 479–488, 2012.

[21] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions-I. *Mathematical Programming*, 14(1):265–294, 1978.

[22] N. Ohsaka, T. Akiba, Y. Yoshida, and K. Kawarabayashi. Fast and accurate influence maximization on large networks with pruned monte-carlo simulations. In *Proc. AAAI*, pages 138–144, 2014.

[23] M. G. Rodriguez, D. Balduzzi, and B. Schölkopf. Uncovering the temporal dynamics of diffusion networks. In *Proc. ICML*, pages 561–568, 2011.

[24] D. Sheldon, B. Dilkina, A. Elmachtoub, R. Finseth, A. Sabharwal, J. Conrad, C. Gomes, D. Shmoys, W. Allen, O. Amundsen, and W. Vaughan. Maximizing the spread of cascades using network design. In *Proc. UAI*, pages 517–526, 2010.

[25] G. Song, X. Zhou, Y. Wang, and K. Xie. Influence maximization on large-scale mobile social network: A divide-and-conquer method. *IEEE Trans. Parallel and Distributed Systems*, 26(5):1379–1392, 2015.

[26] Y. Tang, Y. Shi, and X. Xiao. Influence maximization in near-linear time: A martingale approach. In *Proc. ACM SIGMOD*, pages 1539–1554, 2015.

[27] Y. Tang, X. Xiao, and Y. Shi. Influence maximization: Near-optimal time complexity meets practical efficiency. In *Proc. ACM SIGMOD*, pages 75–86, 2014.

[28] P. Zhang, W. Chen, X. Sun, Y. Wang, and J. Zhang. Minimizing seed set selection with probabilistic coverage guarantee in a social network. In *Proc. KDD*, pages 1306–1315, 2014.

[29] C. Zhou, P. Zhang, J. Guo, X. Zhu, and L. Guo. UBLF: An upper bound based approach to discover influential nodes in social networks. In *Proc. IEEE ICDM*, pages 907–916, 2013.

[30] Y. Zhu, Z. Lu, Y. Bi, W. Wu, Y. Jiang, and D. Li. Influence and profit: Two sides of the coin. In *Proc. IEEE ICDM*, pages 1301–1306, 2013.