Profit Maximization for Viral Marketing in Online Social Networks: Algorithms and Analysis

Jing Tang, *Member, IEEE*, Xueyan Tang, *Member, IEEE*, and Junsong Yuan, *Member, IEEE*

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 selection of initial seed users yields 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. 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. This characteristic differentiates the profit maximization problem from the traditional influence maximization problem. We develop new seed selection algorithms for profit maximization with strong approximation guarantees. We also derive several upper bounds to benchmark the practical performance of an algorithm on any specific problem instance. Experimental evaluations with real OSN datasets demonstrate the effectiveness of our algorithms and techniques.

Index Terms—Online social networks, viral marketing, profit maximization, submodular maximization

1 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 [9].

A large amount of recent work [4], [6], [7], [16], [17], [20], [27], [28], [29], [32], [35], [36], [37], [39] 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. [17] 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 [17], 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 [26]. Many follow-up studies have concentrated on efficient implementation of the algorithm for large-scale OSNs [4], [6], [7], [16], [20], [27], [28], [29], [32], [35], [36], [37], [39].

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. 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. 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. [5]. The original double greedy algorithms

- *J. Tang and X. Tang are with the School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798. E-mail: {tang0311, asxytang}@ntu.edu.sg.*
- *J. Yuan is with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798. E-mail: jsyuan@ntu.edu.sg.*

Manuscript received 30 Nov. 2016; revised 22 Nov. 2017; accepted 23 Dec. 2017. Date of publication 28 Dec. 2017; date of current version 27 Apr. 2018. (Corresponding author: Jing Tang.)
Recommended for acceptance by F. Bonchi.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TKDE.2017.2787757

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 along different directions.

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. To expand the applicability of their approximation guarantees, we develop an iterative pruning technique to provide good warm-starts and relax the condition for the guarantees to hold.
- To further improve the approximation guarantees and deal with cases where the required condition is not fulfilled, we derive several upper bounds on the optimal solution to the profit maximization problem. These bounds can be used to characterize the quality of the solutions constructed 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 2 reviews the related work. Section 3 defines the profit maximization problem. Section 4 elaborates our algorithm design. Section 5 derives the upper bounds. Section 6 presents the experimental study. Finally, Section 7 concludes the paper.

2 Related Work

Influence Maximization. Kempe et al. [17] 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], [20], [35], using heuristics to trade the accuracy of influence estimation for computational efficiency [6], [16], or optimizing the Monte-Carlo simulations for influence estimation [4], [27], [28], [29], [31], [36], [37]. Tang et al. [33] studied algorithms for online processing of influence maximization. There is also some complementary work on learning the parameters of influence propagation models [2], [11], [18], [25]. In addition, some recent work [8], [12], [23], [38] 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. [9] were the first to exploit social influence for marketing optimization by modeling social networks as Markov random fields. Li et al. [22] 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. [13] 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. [1] 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 [24], [40] 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 4.1) and may give poor performance in practice (Section 6.2). 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 [17]. However, the profit function that we define is submodular but not necessarily monotone. There is some work on maximizing non-monotone submodular functions under knapsack constraints [19]. But the proposed algorithms require submodular functions to be non-negative and have very high time complexities (at least $O(n^4 \log n)$ for computation and $O(n^{13})$ for sampling where n is the size of the ground set). Thus, they are not directly applicable to our problem. We do not assume any pre-determined seed set size or seed selection cost. Our profit maximization problem is an unconstrained submodular maximization problem. Feige et al. [10] developed local search algorithms for approximately maximizing non-monotone submodular functions. Recently, Buchbinder et al. [5] 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 4.2). 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 improve our profit maximization solutions and expand the applicability of their approximation guarantees (Section 4.3). Iyer et al. [14], [15] gave two modular upper bounds on submodular functions and made use of them for submodular minimization. Inspired by these studies, we establish several upper bounds on the optimal solution for submodular maximization to benchmark the performance of our algorithms (Section 5).

3 Problem Formulation

3.1 Preliminaries

Let $G = (V, E)$ be a directed graph modeling an OSN, where the nodes V represent users and the edges E represent the connections among users (e.g., friendships on Facebook, followships on Twitter). For each directed edge $\langle u, v\rangle \in E$, we refer to v as a *neighbor* of u, and refer to u as an *inverse neighbor* of v. For ease of reference, Table 1 summarizes some key notations in this paper.

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 using the Independent Cascade (IC) model —a

TABLE 1
Summary of Notations

Notation	Description
$G = (V, E)$	a social graph G with a node set V and an edge set E
N_u	the set of u's neighbors
$V_g(S)$	the nodes activated by S in a sample outcome g
S^*	the optimal seed set
$b(v)$	the benefit generated by activating a node v
$\beta(S)$	the total benefit brought by a seed set S
$c(S), c(v)$	the seed selection cost of a seed set S and a node v
$\phi(S)$	the profit produced by a seed set S, $\phi(S) = \beta(S) - c(S)$

representative and most widely-studied model for influence propagation [6], [7], [16], [17], [20], [27], [29], [32], [36], [37], [39]. In the IC model, a propagation probability $p_{u,v}$ is associated with each edge $\langle u, v\rangle$, representing the probability for v to be activated by u through their connection. Let N_u denote the set of node u's neighbors, i.e., $N_u = \{v : v \in V, \langle u, v\rangle \in E\}$. Given a set of seed nodes S, the IC diffusion process proceeds as follows. Initially, the seed nodes 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 $v \in N_u$, v would become activated with probability $p_{u,v}$. This process repeats until no more node can be activated. Note that the IC diffusion process is a random process. Let $g \sim G$ be a sample outcome of influence propagation in the sample space and let $V_g(S)$ be the set of nodes activated starting from the initial seed set S in the sample outcome g. The *influence spread* of the seed set S, denoted by $\sigma(S)$, is the expected number of nodes activated over all possible sample outcomes of influence propagation, i.e.,

$$\sigma(S) = \mathbb{E}[|V_g(S)|].$$

3.2 The Profit Maximization Problem

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. In general, the users in the social network are likely to bring different amounts of benefit if activated and have different costs for seed selection. For example, users may have different preferences for various models of a product due to their genders, ages, or occupations (e.g., most students may purchase iPhone 7 32 G while most businessmen may purchase iPhone 7 Plus 256 G, thus they offer different benefits), and sending free samples to users in distinct regions may incur different delivery expenses. Suppose that each node $v \in V$ is associated with a benefit $b(v)$ if v is activated and a cost $c(v)$ for seed selection. Then, we define a *profit* metric as the benefit of influence spread less the cost of seed selection, i.e., the profit of a seed set S, denoted by $\phi(S)$, is given by

$$\phi(S) = \beta(S) - c(S),$$

where $\beta(S) = \mathbb{E}[\sum_{v \in V_g(S)} b(v)]$ is the total benefit brought by all the nodes activated and $c(S) = \sum_{v \in S} c(v)$ is the total cost of all the seed nodes selected.

Our goal is to find a seed set S to maximize the profit $\phi(S)$. First, we study the submodularity of the profit function. To simplify the notations, we define $\beta(v \,|\, S) \triangleq \beta(S \cup \{v\}) - \beta(S)$ as the *marginal benefit gain* of adding a new seed node $v \in V$ into a seed set $S \subseteq V$ and define $\phi(v \,|\, S) \triangleq \phi(S \cup \{v\}) - \phi(S)$ as the *marginal profit gain* of adding v into a seed set S.

Proposition 1. *The profit function $\phi(\cdot)$ is submodular if the benefit function $\beta(\cdot)$ is submodular.*

Proof. If $\beta(\cdot)$ is submodular, for any two seed sets S and T where $S \subseteq T$ and any node $v \notin T$, it holds that $\beta(v \,|\, S) \geq \beta(v \,|\, T)$. Therefore, we have $\phi(v \,|\, S) = \beta(v \,|\, S) - c(v) \geq \beta(v \,|\, T) - c(v) = \phi(v \,|\, T)$, which implies that $\phi(\cdot)$ is also submodular. □

Kempe et al. [17] has proved that the influence function $\sigma(\cdot)$ is submodular under the IC model. Using a similar approach, it can be shown that the benefit function $\beta(\cdot)$ is also submodular under the IC model.

Proposition 2. *The benefit function $\beta(\cdot)$ is submodular under the IC model.*

Proof. A sample influence propagation outcome g can be generated by independently flipping a coin of bias $p_{u,v}$ for each edge $\langle u, v\rangle \in E$ to decide whether the edge is *live* or *blocked*. It is easy to see that the set of nodes $V_g(S)$ activated in the sample outcome g are those that can be reached from the seed set S in g. The total benefit generated from the seed set S in the outcome g is given by $\beta_g(S) = \sum_{v \in V_g(S)} b(v)$ Let $p(g)$ denote the probability of a specific outcome g in the sample space. Then, $\beta(S) = \sum_g \big(p(g) \cdot \beta_g(S)\big)$.

In addition, the marginal benefit gain $\beta_g(u \,|\, S) = \sum_{v \in V_g(u \,|\, S)} b(v)$, where $V_g(u \,|\, S)$ is the set of nodes that are reachable from a node u but are not reachable from any node in a seed set S in the sample outcome g. For any two node sets S and T where $S \subseteq T$, we have $V_g(u \,|\, S) \supseteq V_g(u \,|\, T)$, which implies that $\beta_g(u \,|\, S) \geq \beta_g(u \,|\, T)$. Since $p(g) \geq 0$ for any g, taking the linear combination, we have $\beta(u \,|\, S) \geq \beta(u \,|\, T)$. Thus, $\beta(\cdot)$ is submodular. □

By Propositions 1 and 2, the profit function $\phi(\cdot)$ is also submodular under the IC model. Though both the profit and influence 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. The marginal profit gain by adding a new seed, i.e., $\phi(v \,|\, S) = \beta(v \,|\, S) - c(v)$, can be negative. As shall be shown soon, this makes the seed selection for profit maximization more challenging than that for influence maximization. Selecting seed nodes to maximize the profit becomes an unconstrained submodular maximization problem [5], [10].

4 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 [5], and develop new methods for profit maximization.

4.1 Simple Greedy Heuristic

If activating each node offers the same benefit, one may intuitively 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 $|V|$) and then chooses the best solution. But unfortunately, this would not work when the nodes have different costs for seed selection because

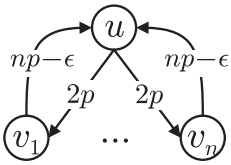


Fig. 1. A simple hill-climbing algorithm fails to achieve any bounded approximation factor.

most existing influence maximization algorithms do not differentiate the nodes by their costs in the seed selection. If a top influential node has a high cost (e.g., a popular user may require more incentives to be recruited as a seed), the total profit of the seed set selected can be low or even negative.

We construct a simple greedy hill-climbing algorithm to optimize the profit, similar to that proposed by Kempe et al. [17] for influence maximization. Algorithm 1 describes the greedy heuristic. It starts with an empty seed set $S = \emptyset$. In each iteration, if the largest marginal profit gain $\phi(v \mid S)$ by choosing a new seed from the non-seed nodes $V \setminus S$ is positive, the greedy heuristic adds the corresponding node to 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 S. This simple greedy algorithm shares the same spirit with the hill-climbing heuristic adopted by [24] and [40].

Algorithm 1. SimpleGreedy(G, ϕ)

1: initialize $S \leftarrow \emptyset$;
2: **while** True **do**
3: find $u \leftarrow \arg\max_{v \in V \setminus S}\{\phi(v \mid S)\}$;
4: **if** $\phi(u \mid S) \leq 0$ **then**
5: break;
6: $S \leftarrow S \cup \{u\}$;
7: **return** S;

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 $c(v) = 1$ for seed selection and the same unit benefits $b(v) = 1$ when activated. Fig. 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 $\phi(v_i \mid \{u\}) = 2\ (n-1)p - 2np = -2p < 0$. Therefore, the greedy algorithm would stop after selecting u and return $S = \{u\}$ with a profit of $\phi(\{u\}) = 2np$. On the other hand, when nodes $v_1, v_2, , v_n$ are all chosen as seeds, the probability for node u to be activated is $1 - (1 - np + \epsilon)^n$. As a result, $\phi(V \setminus \{u\}) = n + 1 - \ (1 - np + \epsilon)^n - n = 1 - (1 - np + \epsilon)^n$. 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 $\phi(V \setminus \{u\}) = 1 - (1 - \frac{1}{n} + \frac{1}{4n^2})^n = 1 - \ (1 - \frac{1}{2n})^{2n}$. 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 [12], [23], [38] for achieving a given amount of influence spread would not be able to provide strong approximation guarantees for profit maximization either. As analyzed in [12], [38], for any $\epsilon > 0$, the seed minimization problem cannot be approximated within a factor of $(1 - \epsilon) \ln |V|$ 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.

4.2 Double Greedy Algorithms

Buchbinder et al. [5] 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 S and a set 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 S and T. When the algorithms complete, it must hold that $S = T$ 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 S (i.e., $\phi(S \cup \{u\}) - \phi(S) = \phi(u \mid S)$) and the marginal profit gain of removing u from T (i.e., $\phi(T \setminus \{u\}) - \phi(T) = -\phi(u \mid T \setminus \{u\})$). In the deterministic approach, each node u joins S if it generates higher marginal profit gain than that of quitting from T and vice versa (lines 5–8). In the randomized approach, each node u is added to S with probability $\phi(u \mid S)/\big(\phi(u \mid S) - \phi(u \mid T \setminus \{u\})\big)$, and is removed from T with probability $-\phi(u \mid T \setminus \{u\})/\big(\phi(u \mid S) - \phi(u \mid T \setminus \{u\})\big)$.

In general, in the double greedy algorithms, if we initialize S with S_0 and T with T_0 where $\emptyset \subseteq S_0 \subseteq T_0 \subseteq V$ (line 1 in Algorithm 2) and only check the nodes in $T_0 \setminus S_0$ to decide whether or not to include them in S and T (line 2), we have the following proposition according to [5].[1]

Proposition 3. *Let S^* denote the optimal solution such that $\phi(S^*) = \max_{S \subseteq V} \phi(S)$. If S and T are initialized with S_0 and T_0 respectively, Algorithm 2 returns a solution S_D satisfying*

$$\phi\big((S^* \cup S_0) \cap T_0\big) + \phi(S_0) + \phi(T_0) \leq 3 \cdot \phi(S_D),$$

1. Proposition 3 is extracted through a detailed check of the proof of Theorem I.1 in [5] though it was not presented as a separate proposition therein.

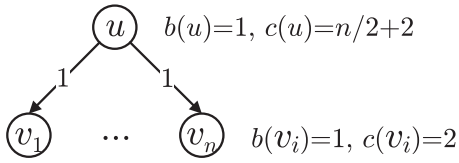


Fig. 2. Double greedy algorithms fail to achieve any bounded approximation factor when $\phi(V) < 0$.

and the randomized version of Algorithm 2 returns a solution S_R satisfying

$$\phi\big((S^* \cup S_0) \cap T_0\big) + \big(\phi(S_0) + \phi(T_0)\big)/2 \leq 2 \cdot \mathbb{E}[\phi(S_R)].$$

Algorithm 2. DeterministicDoubleGreedy(G, ϕ)[5]

1: start with $S \leftarrow \emptyset, T \leftarrow V$;
2: **for** each node $u \in V$ **do**
3: $r^+ \leftarrow \phi(u \,|\, S)$;
4: $r^- \leftarrow -\phi(u \,|\, T \setminus \{u\})$;
5: **if** $r^+ \geq r^-$ **then**
6: $S \leftarrow S \cup \{u\}$;
7: **else**
8: $T \leftarrow T \setminus \{u\}$;
9: **return** S $(= T)$;
// 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 $r^+ + r^- = 0$.

Proposition 3 gives rise to the approximation guarantees proved in [5].

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

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

and the expected profit of the seed set S_R returned by the randomized version of Algorithm 2 satisfies

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

Remark. Buchbinder et al. [5] established the above approximation guarantees for any non-negative submodular function $\phi(\cdot)$ when S and T are initialized with $\emptyset$ and V in the double greedy algorithms. In this case, since $(S^* \cup S_0) \cap T_0 = (S^* \cup \emptyset) \cap V = S^* \cap V = S^*$, by Proposition 3, it follows that $\phi(S^*) + \phi(\emptyset) + \phi(V) \leq 3 \cdot \phi(S_D)$ and $\phi(S^*) + \big(\phi(\emptyset) + \phi(V)\big)/2 \leq 2 \cdot \mathbb{E}[\phi(S_R)]$. 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 $\phi(\emptyset) + \phi(V) \geq 0$ is required to provide the approximation guarantees. In our profit maximization problem, since $\phi(\emptyset) = 0$, we just need the condition $\phi(V) \geq 0$.

4.3 Warm-Start by Iterative Pruning

Theorem 1 provides strong theoretical guarantees for the approximability of Algorithm 2. However, the condition of $\phi(V) \geq 0$ may not be realistic in our profit maximization problem. $\phi(V) \geq 0$ 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 $\phi(V) < 0$, we cannot have any bounded approximation guarantees by directly applying Algorithm 2. For example, consider the social network in Fig. 2 with $n+1$ nodes ($n \geq 3$) and n edges. Let propagation probabilities $p_{u,v_i} = 1$, and let benefits $b(u) = 1$ and $b(v_i) = 1$ for each v_i. Let seed selection costs $c(u) = \frac{n}{2} + 2$ and $c(v_i) = 2$ for each v_i. Then, $\phi(V) = (n+1) - (\frac{n}{2} + 2 + 2n) = -\frac{3n}{2} - 1 < 0$. Assume that Algorithm 2 initializes $S = \emptyset$ and $T = V$, and it iterates through the nodes in the order of $u, v_1, v_2, \ldots, v_n$. In the first iteration, we have $\phi(u \,|\, S) = \phi(\{u\}) = \beta(\{u\}) - c(u) = n + 1 - (\frac{n}{2} + 2) = \frac{n}{2} - 1$. Meanwhile, $-\phi(u \,|\, T \setminus \{u\}) = -\big(\beta(V) - \beta(V \setminus \{u\}) - c(u)\big) = -\big((n+1) - n - (\frac{n}{2} + 2)\big) = \frac{n}{2} + 1$. Thus, u quits from T so that $S = \emptyset$ and $T = V \setminus \{u\}$. In the second iteration, $\phi(v_1 \,|\, S) = \beta(\{v_1\}) - \beta(\emptyset) - c(v_1) = 1 - 2 = -1$ and $-\phi(v_1 \,|\, T \setminus \{v_1\}) = -\big(\beta(V \setminus \{u\}) - \beta(V \setminus \{u, v_1\}) - c(v_1)\big) = -\big(n - (n-1) - 2\big) = 1$. By Algorithm 2, v_1 quits from T as well. Similarly, in each subsequent iteration, v_i quits from T. Thus, Algorithm 2 finally returns $S = \emptyset$ with profit $\phi(S) = 0$. 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 $\phi(V) < 0$.

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. Our strategy is 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. Given the profit function $\phi(\cdot)$, we define two node sets $A_1 = \{v : \phi(v \,|\, V \setminus \{v\}) > 0\}$ and $B_1 = \{v : \phi(v \,|\, \emptyset) \geq 0)\}$. Due to the submodularity of $\phi(\cdot)$, we have $A_1 \subseteq B_1$ and this allows us to define a space $\Omega_1 = [A_1, B_1]$ that contains all the sets S satisfying $A_1 \subseteq S \subseteq B_1$.

Proposition 4. *$\Omega_1 = [A_1, B_1]$ retains all global maximizers S^* for the profit function $\phi(\cdot)$, i.e., $A_1 \subseteq S^* \subseteq B_1$ for all S^* where $\phi(S^*) = \max_{S \subseteq V} \phi(S)$.*

Proof. If $\phi(v \,|\, V \setminus \{v\}) > 0$, for any seed set $S \subseteq V \setminus \{v\}$, it follows from the submodularity of $\phi(\cdot)$ that $\phi(v \,|\, S) \geq \phi(v \,|\, V \setminus \{v\}) > 0$. Thus, $S \cup \{v\}$ always generates higher profit than S, so v must be selected as a seed in every optimal solution, which indicates $A_1 \subseteq S^*$. By similar arguments, if $\phi(v \,|\, \emptyset) < 0$, then v cannot be selected as a seed in any optimal solution, which implies $S^* \subseteq B_1$. □

We can prune $\Omega_1 = [A_1, B_1]$ even further using an iterative strategy. Specifically, since the nodes in A_1 must be included in any global maximizer, we can shrink B_1 to $B_2 = \{v : \phi(v \,|\, A_1) \geq 0\}$. Similarly, since the nodes in $V \setminus B_1$ cannot be included in any global maximizer, we can expand A_1 to $A_2 = \{v : \phi(v \,|\, B_1 \setminus \{v\}) > 0\}$. This yields a smaller space $\Omega_2 = [A_2, B_2]$ than Ω_1. These operations can be repeated alternately until A and B cannot be further broadened and narrowed respectively. Algorithm 4.3 presents the pseudo code of the iterative pruning process. Let A^* and B^* denote the node sets finally returned by Algorithm 4.3. Theorem 2 proves that $\Omega^* = [A^*, B^*]$ retains all global maximizers. For notational convenience, we define $A_0 = \emptyset$ and

$B_0 = V$. To establish Theorem 2, we also make use of the following proposition about submodular functions.

Algorithm 3. IterativePrune(G, ϕ)

1: start with $t = 0$, $A_0 \leftarrow \emptyset$, $B_0 \leftarrow V$;
2: **repeat**
3: $A_{t+1} \leftarrow \{v : \phi(v \mid B_t \setminus \{v\}) > 0)\}$;
4: $B_{t+1} \leftarrow \{v : \phi(v \mid A_t) \geq 0)\}$;
5: $t \leftarrow t + 1$;
6: **until** *converged, i.e.,* $A_t = A_{t-1}$ *and* $B_t = B_{t-1}$;
7: **return** A_t and B_t;

Proposition 5 ([26]). *For any submodular function $\phi(\cdot)$ on the power set of V and any two subsets $X, Y \subseteq V$, it holds that*

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

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

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

Proof. We first show that after each iteration, the newly generated space is reduced from that in the previous iteration, i.e., $A_t \subseteq A_{t+1} \subseteq B_{t+1} \subseteq B_t$. We prove it by induction. Obviously, $A_0 = \emptyset \subseteq A_1 \subseteq B_1 \subseteq V = B_0$ according to Proposition 4. Suppose that $A_{t-1} \subseteq A_t \subseteq B_t \subseteq B_{t-1}$ holds for some $t \geq 1$. For every node $v \in A_t$, we know $\phi(v \mid B_{t-1} \setminus \{v\}) > 0$. Due to the submodularity, we have $\phi(v \mid B_t \setminus \{v\}) \geq \phi(v \mid B_{t-1} \setminus \{v\}) > 0$. As a result, $A_t \subseteq A_{t+1}$. Similarly, for every node $v \in B_{t+1}$, we have $\phi(v \mid A_t) \geq 0$. Due to the submodularity, $\phi(v \mid A_{t-1}) \geq \phi(v \mid A_t) \geq 0$, which indicates that $B_{t+1} \subseteq B_t$. Furthermore, for all nodes $v \in A_{t+1} \cap A_t$, we have $\phi(v \mid A_t) = 0$, which implies that $(A_{t+1} \cap A_t) \subseteq B_{t+1}$ via line 4 in Algorithm 4.3. For all nodes $v \in A_{t+1} \setminus A_t$, we have $\phi(v \mid B_t \setminus \{v\}) > 0$. Since $v \notin A_t$ and $A_t \subseteq B_t$, we also have $A_t \subseteq B_t \setminus \{v\}$. Thus, $\phi(v \mid A_t) \geq \phi(v \mid B_t \setminus \{v\}) > 0$, which implies that $(A_{t+1} \setminus A_t) \subseteq B_{t+1}$. Consequently, we have $A_{t+1} = (A_{t+1} \cap A_t) \cup (A_{t+1} \setminus A_t) \subseteq B_{t+1}$. Therefore, $A_t \subseteq A_{t+1} \subseteq B_{t+1} \subseteq B_t$ holds for any $t \geq 0$.

Next, we explore the relationship of S^* to A^* and B^*. Obviously, $A_0 = \emptyset \subseteq S^* \subseteq V = B_0$ holds. As proved in Proposition 4, $A_1 \subseteq S^* \subseteq B_1$ also holds. Suppose that $A_t \subseteq S^* \subseteq B_t$ holds for some $t \geq 0$. Then, any node v satisfying $\phi(v \mid B_t \setminus \{v\}) > 0$ must be in S^*. Otherwise, if $v \notin S^*$, we have $\phi(v \mid S^*) = \phi(v \mid S^* \setminus \{v\}) \geq \phi(v \mid B_t \setminus \{v\}) > 0$ by the submodularity, which indicates $\phi(S^* \cup \{v\}) > \phi(S^*)$, contradicting the optimality of S^*. The set of such nodes v satisfying $\phi(v \mid B_t \setminus \{v\}) > 0$ is exactly A_{t+1}, and hence $A_{t+1} \subseteq S^*$. By induction, we have $A_t \subseteq S^* \subseteq B_t$ for any $t \geq 0$ and thus, $A^* \subseteq S^* \subseteq B^*$.

Finally, we show that $\phi(A_t) \leq \phi(A_{t+1})$ and $\phi(B_t) \leq \phi(B_{t+1})$ for any $t \geq 0$. In fact, for any node $v \in A_{t+1} \setminus A_t$, it holds that $\phi(v \mid A_{t+1} \setminus \{v\}) \geq \phi(v \mid B_{t+1} \setminus \{v\}) \geq \phi(v \mid B_t \setminus \{v\}) > 0$, where the first two inequalities are due to the submodularity (since $A_{t+1} \subseteq B_{t+1} \subseteq B_t$) and the third inequality is by the definition of A_{t+1}. Therefore, $\phi(A_{t+1}) \geq \phi(A_t) + \sum_{v \in A_{t+1} \setminus A_t} \phi(v \mid A_{t+1} \setminus \{v\}) \geq \phi(A_t)$, where the first inequality is due to (1) of Proposition 5 and the fact $A_t \subseteq A_{t+1}$. Similarly, it can be shown that $\phi(B_t) \leq \phi(B_{t+1})$. This completes the proof. □

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

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

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

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

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

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

Corollary 1 shows that conducting the iterative pruning prior to applying the double greedy algorithms allows us to maintain the same approximation guarantees with the condition $\phi(A^*) + \phi(B^*) \geq 0$. This condition is much weaker than the original condition $\phi(V) \geq 0$ of Theorem 1 since by Theorem 2, $\phi(V) = \phi(\emptyset) + \phi(V) = \phi(A_0) + \phi(B_0) \leq \phi(A_1) + \phi(B_1) \leq \cdots \leq \phi(A^*) + \phi(B^*)$. Thus, Corollary 1 significantly expands the applicability of the theoretical guarantees.

For the example shown in Fig. 2, $\phi(u \mid V \setminus \{u\}) = 1 - (\frac{n}{2} + 2) < 0$ and $\phi(v_i \mid V \setminus \{v_i\}) = 0 - 2 < 0$ for each $1 \leq i \leq n$. Thus, by Algorithm 4.3, $A_1 = \emptyset$. Furthermore, $\phi(u \mid \emptyset) = n + 1 - (\frac{n}{2} + 1) = \frac{n}{2} - 1 > 0$ and $\phi(v_i \mid \emptyset) = 1 - 2 = -1 < 0$ for each $1 \leq i \leq n$, which implies $B_1 = \{u\}$. In the second iteration, since $\phi(u \mid B_1 \setminus \{u\}) = \phi(u \mid \emptyset) = \frac{n}{2} - 1 > 0$, we have $A_2 = \{u\}$. B_2 remains the same as $B_1 = \{u\}$. Thus, Algorithm 4.3 returns $A^* = B^* = \{u\}$. As a result, $\phi(A^*) + \phi(B^*) = 2 \cdot \phi(\{u\}) = n - 2 \geq 0$ satisfying the condition given in Corollary 1. In fact, since $A^* = B^*$, it must be an optimal solution and the double greedy algorithms return exactly the optimal solution.

Finally, we remark that the sets A^* and B^* produced by the iterative pruning can also be used as warm-starts for the simple greedy algorithm in Section 4.1 so that only the nodes in $B^* \setminus A^*$ need to be further examined for seed selection.

4.4 Generality and Time Complexity

Our analysis and algorithms are general frameworks that can be adapted to any influence propagation models which are submodular. These include the IC model, the LT model, the generalized triggering model [17], continuous-time models [30], and topic-aware models [3], to name a few.

Evaluating the profit metric involves estimating the influence spread given a seed set. Any existing influence estimation methods, such as Monte-Carlo simulation [17], [20], [29], [31] and Reverse Influence Sampling (RIS) [4], [27], [28], [36],

[37], 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 a node u into a seed set S or removing a node u from a seed set T is $O(M)$. The simple greedy algorithm (Algorithm 1) takes at most $O\big((|V|-|S|)M\big)$ time to select one seed. Thus, the total time complexity of the simple greedy algorithm is $O\big((|V|+|V|-1++1)M\big) = O(|V|^2M)$. 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 $O(|V|\cdot 2M) = O(|V|M)$. For the iterative pruning process (Algorithm 4.3), the size of the node set $B_t \setminus A_t$ to check reduces by at least 1 in each iteration. Therefore, it has a time complexity of $O(|V|^2M)$.

5 Analysis of Upper Bounds

The previous section has established a uniform theoretical guarantee across all instances of the profit maximization problem under a certain condition. To further improve the approximation guarantees for specific problem instances and to deal with the instances where the required condition is not satisfied, we next derive several upper bounds on the optimal solution to the profit maximization problem. These bounds are easy to compute and can be used to characterize the quality of the solutions constructed on any problem instances.

5.1 Upper Bound for Double Greedy

For the deterministic double greedy algorithm (Algorithm 2), besides the approximation guarantee provided by Corollary 1, we can also derive an upper bound on the optimal solution without any condition requirement. In fact, following the proof of Corollary 1, we have an upper bound

$$\mu_1 \triangleq 3\cdot\phi(\hat{S}_D) - \big(\phi(A^*)+\phi(B^*)\big)$$

on the profit of the optimal solution $\phi(S^*)$. Regardless of whether $\phi(A^*)+\phi(B^*)$ is non-negative or not, the above upper bound always holds. Thus, any seed set $\hat{S}_D$ returned by the double greedy algorithm has an approximation guarantee of $\phi(\hat{S}_D)/\mu_1$. When $\phi(A^*)+\phi(B^*) > 0$, this guarantee is tighter than the constant factor $(1/3)$ of Corollary 1.

5.2 Upper Bounds Based on Submodularity

Now, we develop several upper bounds based solely on the submodularity of the profit function. In a nutshell, based on a given node set X, we first derive two modular upper bounds $m_X(Y)$ and $\bar{m}_X(Y)$ on $\phi(Y)$ for any node set Y. Then, we find the maxima of $m_X(Y)$ and $\bar{m}_X(Y)$ over all sets Y to derive two upper bounds $\mu(X)$ and $\bar{\mu}(X)$ on the maximum achievable profit $\max_{S\subseteq V}\phi(S)$. Specifically, leveraging Proposition 5 and restricting the sets X and Y within the boundaries A^* and B^* discovered by the iterative pruning (i.e., $A^*\subseteq X, Y\subseteq B^*$), we can obtain two upper bounds on $\phi(Y)$ as follows:

$$m_X(Y) \triangleq \phi(X) - \sum_{v\in X\setminus Y}\phi(v\,|\,B^*\setminus\{v\}) + \sum_{v\in Y\setminus X}\phi(v\,|\,X), \quad (3)$$

$$\bar{m}_X(Y) \triangleq \phi(X) - \sum_{v\in X\setminus Y}\phi(v\,|\,X\setminus\{v\}) + \sum_{v\in Y\setminus X}\phi(v\,|\,A^*). \quad (4)$$

Proposition 6. *For any two sets X and Y where $A^*\subseteq X$, $Y\subseteq B^*$, the above defined $m_X(Y)$ and $\bar{m}_X(Y)$ satisfy*

$$m_X(Y)\geq\phi(Y) \quad \text{and} \quad \bar{m}_X(Y)\geq\phi(Y).$$

Proof. For each node $v\in X\setminus Y$, since $X\cup Y\setminus\{v\}\subseteq B^*\setminus\{v\}$, it holds that $\phi(v\,|\,B^*\setminus\{v\})\leq\phi(v\,|\,X\cup Y\setminus\{v\})$ due to the submodularity. By (1) and (3), we obtain $m_X(Y)\geq\phi(Y)$. Similarly, for each node $v\in Y\setminus X$, since $A^*\subseteq X\cap Y$, it holds that $\phi(v\,|\,A^*)\geq\phi(v\,|\,X\cap Y)$. By (2) and (4), we obtain $\bar{m}_X(Y)\geq\phi(Y)$. □

Based on Proposition 6, we can find two series of upper bounds on the maximum value of $\phi(\cdot)$ as follows. For any set X where $A^*\subseteq X\subseteq B^*$, we define

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

Theorem 3. *For any set X where $A^*\subseteq X\subseteq B^*$,*

$$\mu(X)\geq\max_{S\subseteq V}\phi(S) \quad \text{and} \quad \bar{\mu}(X)\geq\max_{S\subseteq V}\phi(S).$$

Proof. In fact, $\mu(X)=\max_{A^*\subseteq Y\subseteq B^*}m_X(Y)\geq\max_{A^*\subseteq Y\subseteq B^*}\phi(Y)=\max_{S\subseteq V}\phi(S)$, where the first equality is by the definition of $\mu(X)$, the second inequality is due to Proposition 6, and the last equality is due to Theorem 2. Similarly, we have $\bar{\mu}(X)\geq\max_{S\subseteq V}\phi(S)$. □

A nice feature of the above upper bounds is that $m_X(Y)$ and $\bar{m}_X(Y)$ are modular functions with respect to Y. A function $m(\cdot)$ is modular iff for any node v and any two sets S and T, $m(v\,|\,S)=m(v\,|\,T)$ where $m(v\,|\,S)=m(S\cup\{v\})-m(S)$, or equivalently, $m(S)=m(\emptyset)+\sum_{v\in S}m(v\,|\,\emptyset)$. It is easy to verify that

$$m_X(Y)=m_X(\emptyset)+\sum_{v\in Y}\big(m_X(\{v\})-m_X(\emptyset)\big).$$

Thus, the upper bound $\mu(X)$ is given by

$$\mu(X)=m_X(\emptyset)+\sum_{v\in B^*}\max\big\{0,m_X(\{v\})-m_X(\emptyset)\big\}. \quad (5)$$

Meanwhile, it is easy to show that

$$m_X(\{v\})-m_X(\emptyset)=\begin{cases}\phi(v\,|\,B^*\setminus\{v\}), & \text{if } v\in X,\\ \phi(v\,|\,X), & \text{otherwise.}\end{cases} \quad (6)$$

Therefore, given X, $\mu(X)$ can be computed in $O(|B^*|)$ time. Similarly, $\bar{\mu}(X)$ can also be computed in $O(|B^*|)$ time. Interested readers are referred to the appendix in the supplementary file for details, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TKDE.2017.2787757.

Remark. Note that when the iterative pruning is not applied, B^* should be replaced by V in (3) and A^* should be replaced by $\emptyset$ in (4). We refer to these corresponding upper bounds on $\phi(Y)$ as $m'_X(Y)$ and $\bar{m}'_X(Y)$. Due to the submodularity of the profit function $\phi(\cdot)$, we have $\phi(v\,|\,B^*\setminus\{v\})\geq\phi(v\,|\,V\setminus\{v\})$ and $\phi(v\,|\,A^*)\leq\phi(v\,|\,\emptyset)$ for any node v. Thus, it holds that $m_X(Y)\leq m'_X(Y)$ and $\bar{m}_X(Y)\leq\bar{m}'_X(Y)$. Therefore, an additional advantage of the iterative pruning is to make the upper bounds tighter.

5.3 Tighten the Bounds

Theorem 3 indicates that we can obtain upper bounds on the optimal profit by choosing any set X in $\mu(X)$ and $\bar{\mu}(X)$. However, it is not known yet what kind of X leads to tighter bounds. In the following, we show that the seed set returned by a greedy algorithm (either simple greedy or double greedy algorithms) can provide the *tightest* bound among all the seed sets examined by the greedy algorithm. To prove this claim, we start by showing that the second order marginal profit gain is diminishing if the set of nodes activated by a seed set S is a union of the nodes activated by each individual seed $u \in S$. Recall that $V_g(S)$ denotes the set of nodes activated by a seed set S in a sample outcome X of influence propagation.

Proposition 7. *If $V_g(S) = \bigcup_{u\in S} V_g(\{u\})$ holds for any set S under an influence propagation model, then we have*

$$\phi(v \mid S) - \phi(v \mid T) \geq \phi(v \mid S \cup R) - \phi(v \mid T \cup R),$$

for any sets R, S, T where $S \subseteq T$, and any node $v \notin T \cup R$.

Proof. We know that $\phi(v \mid S) = \beta(v \mid S) - c(v)$ for any node $v \notin S$. So, for any node $v \notin T \supseteq S$,

$$\phi(v \mid S) - \phi(v \mid T) = \beta(v \mid S) - \beta(v \mid T). \tag{7}$$

On the other hand, in any sample outcome X of influence propagation, $V_g(S) \subseteq V_g(T)$ when $S \subseteq T$, which indicates $V_g(v \mid S) \supseteq V_g(v \mid T)$ where $V_g(v \mid S) \triangleq V_g(S \cup \{v\}) \setminus V_g(S)$ is the set of nodes activable by u but not S. Thus, by definition,

$$\begin{aligned}\beta(v \mid S) - \beta(v \mid T) &= \mathbb{E}\Big[\sum_{u\in V_g(v\mid S)} b(u)\Big] - \mathbb{E}\Big[\sum_{u\in V_g(v\mid T)} b(u)\Big] \\ &= \mathbb{E}\Big[\sum_{u\in V_g(v\mid S)\setminus V_g(v\mid T)} b(u)\Big].\end{aligned}$$

As a result, (7) can be rewritten as

$$\phi(v \mid S) - \phi(v \mid T) = \mathbb{E}\Big[\sum_{u\in V_g(v\mid S)\setminus V_g(v\mid T)} b(u)\Big].$$

Similarly, we have

$$\phi(v \mid S \cup R) - \phi(v \mid T \cup R) = \mathbb{E}\Big[\sum_{u\in V_g(v\mid S\cup R)\setminus V_g(v\mid T\cup R)} b(u)\Big].$$

Thus, the proposition holds if $V_g(v \mid S) \setminus V_g(v \mid T) \supseteq V_g(v \mid S \cup R) \setminus V_g(v \mid T \cup R)$. We prove this relation as follows. First, since $V_g(S) = \bigcup_{u\in S} V_g(\{u\})$, we have $V_g(v \mid S) = \big(\bigcup_{u\in S\cup\{v\}} V_g(\{u\})\big) \setminus \big(\bigcup_{u\in S} V_g(\{u\})\big) = V_g(\{v\}) \setminus V_g(S)$. Therefore,

$$\begin{aligned}&V_g(v \mid S) \setminus V_g(v \mid T) \\ &\quad= \Big(V_g(\{v\}) \setminus V_g(S)\Big) \setminus \Big(V_g(\{v\}) \setminus V_g(T)\Big) \\ &\quad= \Big(V_g(\{v\}) \setminus V_g(S) \setminus V_g(\{v\})\Big) \\ &\qquad\cup \Big(V_g(\{v\}) \setminus V_g(S) \cap V_g(T)\Big) \\ &\quad= V_g(\{v\}) \setminus V_g(S) \cap V_g(T) \\ &\quad= V_g(\{v\}) \cap V_g(T) \setminus V_g(S).\end{aligned}$$

Similarly,

$$\begin{aligned}&V_g(v \mid S \cup R) \setminus V_g(v \mid T \cup R) \\ &\quad= V_g(\{v\}) \cap V_g(T \cup R) \setminus V_g(S \cup R) \\ &\quad= V_g(\{v\}) \cap V_g(T) \setminus V_g(S \cup R).\end{aligned}$$

Since $V_g(S) \subseteq V_g(S \cup R)$, the target relation is proven. □

Many influence propagation models, including the IC and LT models, fulfill the condition $V_g(S) = \bigcup_{u\in S} V_g(\{u\})$ in Proposition 7. By the submodularity of $\phi(\cdot)$, we already know that $\phi(v \mid S)$ decreases with the base set S. Proposition 7 shows that the decreasing rate is slowing down as the base set expands. This implies that the marginal profit gain drops faster in the earlier stage of seed selection than in the later stage. Based on Proposition 7, we can further establish the monotonicity of the upper bounds $\mu(X)$ and $\bar{\mu}(X)$ as X expands.

Proposition 8. *For any set X where $A^* \subseteq X \subseteq B^*$, and any node $u \in B^* \setminus A^*$, we have*

$$\begin{cases}\mu(X) \geq \mu(X \setminus \{u\}) & \text{if} \quad \phi(X) \leq \phi(X \setminus \{u\}), \\ \bar{\mu}(X) \geq \bar{\mu}(X \cup \{u\}) & \text{if} \quad \phi(X) \leq \phi(X \cup \{u\}).\end{cases}$$

Proof. Based on (5) and (6) in the earlier discussion, we can rewrite the upper bound $\mu(X)$ as

$$\begin{aligned}\mu(X) = m_X(\emptyset) &+ \sum_{v\in X} \max\{0, \phi(v \mid B^* \setminus \{v\})\} \\ &+ \sum_{v\in B^*\setminus X} \max\{0, \phi(v \mid X)\}.\end{aligned}$$

Similarly, for any node $u \in X \setminus A^*$,

$$\begin{aligned}\mu(X \setminus \{u\}) = m_{X\setminus\{u\}}(\emptyset) &+ \sum_{v\in(X\setminus\{u\})} \max\{0, \phi(v \mid B^* \setminus \{v\})\} \\ &+ \sum_{v\in B^*\setminus(X\setminus\{u\})} \max\{0, \phi(v \mid X \setminus \{u\})\}.\end{aligned}$$

It is easy to show that

$$m_X(\emptyset) - m_{X\setminus\{u\}}(\emptyset) = \phi(u \mid X \setminus \{u\}) - \phi(u \mid B^* \setminus \{u\}).$$

As a result,

$$\begin{aligned}&\mu(X) - \mu(X \setminus \{u\}) \\ &\quad= \phi(u \mid X \setminus \{u\}) + \max\{0, -\phi(u \mid B^* \setminus \{u\})\} \\ &\qquad+ \sum_{v\in B^*\setminus X} \Big(\max\{0, \phi(v \mid X)\} - \max\{0, \phi(v \mid X \setminus \{u\})\}\Big) \\ &\qquad- \max\{0, \phi(u \mid X \setminus \{u\})\}.\end{aligned}$$

If $\phi(X) \leq \phi(X \setminus \{u\})$, we have $\phi(u \mid B^* \setminus \{u\}) \leq \phi(u \mid X \setminus \{u\}) \leq 0$. Thus, $\max\{0, -\phi(u \mid B^* \setminus \{u\})\} = -\phi(u \mid B^* \setminus \{u\})$ and $\max\{0, \phi(u \mid X \setminus \{u\})\} = 0$. On the other hand, for any node $v \in B^* \setminus X$, we have $\max\{0, \phi(v \mid X)\} - \phi(v \mid X) = \max\{-\phi(v \mid X), 0\} \geq \max\{-\phi(v \mid X \setminus \{u\}), 0\} = \max\{0, \phi(v \mid X \setminus \{u\})\} - \phi(v \mid X \setminus \{u\})$. That is, $\max\{0, \phi(v \mid X)\} - \max\{0, \phi(v \mid X \setminus \{u\})\} \geq \phi(v \mid X) - \phi(v \mid X \setminus \{u\}) = \phi(X \cup \{v\}) - \phi(X) - \phi(X \cup \{v\} \setminus \{u\}) + \phi(X \setminus \{u\}) = \phi(u \mid X \cup \{v\} \setminus \{u\}) - \phi(u \mid X \setminus \{u\})$. Therefore,

$$\begin{aligned}&\mu(X) - \mu(X \setminus \{u\})\\ &\quad \geq \phi(u \mid X \setminus \{u\}) - \phi(u \mid B^* \setminus \{u\})\\ &\qquad + \sum_{v \in B^* \setminus X} \Big(\phi(u \mid X \cup \{v\} \setminus \{u\}) - \phi(u \mid X \setminus \{u\})\Big).\end{aligned}$$

Let $v_1, v_2, \ldots, v_i$ be the set of elements in $B^* \setminus X$. Then,

$$\begin{aligned}&\sum_{v \in B^* \setminus X} \Big(\phi(u \mid X \cup \{v\} \setminus \{u\}) - \phi(u \mid X \setminus \{u\})\Big)\\ &\quad = \sum_{j=1}^{i} \Big(\phi(u \mid X \cup \{v_j\} \setminus \{u\}) - \phi(u \mid X \setminus \{u\})\Big)\\ &\quad \geq \sum_{j=1}^{i} \Big(\phi(u \mid X \cup \{v_1, \ldots, v_j\} \setminus \{u\})\\ &\qquad\qquad - \phi(u \mid X \cup \{v_1, \ldots, v_{j-1}\} \setminus \{u\})\Big)\\ &\quad = \phi(u \mid B^* \setminus \{u\}) - \phi(u \mid X \setminus \{u\}).\end{aligned}$$

where the inequality is due to Proposition 7. Hence, we can conclude that $\mu(X) - \mu(X \setminus \{u\}) \geq 0$ if $\phi(X) \leq \phi(X \setminus \{u\})$.

The proof of $\bar{\mu}(X) \geq \bar{\mu}(X \cup \{u\})$ if $\phi(X) \leq \phi(X \cup \{u\})$ is analogous. □

Proposition 8 indicates that the higher the profit of a seed set, the tighter the corresponding upper bound derived based on the seed set. Intuitively, the seed sets constructed by the greedy algorithms have increasing profits over iterations and thus, the final seed sets returned by the greedy algorithms can provide the tightest bounds. In the following, we prove it formally. Recall that the greedy algorithms start with the boundary node sets A^* and B^* discovered by the iterative pruning.

Theorem 4. *For the seed set S^g returned by the simple greedy algorithm or the double greedy algorithms, we have*

$$\begin{aligned}&\mu(S^g) \leq \mu(X_B) \leq \mu(B^*) = \bar{\mu}(B^*),\\ \text{and}\quad &\bar{\mu}(S^g) \leq \bar{\mu}(X_A) \leq \bar{\mu}(A^*) = \mu(A^*),\end{aligned}$$

where X_B is any intermediate seed set from B^* to S^g and X_A is any intermediate seed set from A^* to S^g in the execution of the greedy algorithm.

Proof. For any set Y where $A^* \subseteq Y \subseteq B^*$, we have $A^* \setminus Y = \emptyset$. Thus, it follows from (3) and (4) that

$$m_{A^*}(Y) = \phi(A^*) + \sum_{v \in Y \setminus A^*} \phi(v \mid A^*) = \bar{m}_{A^*}(Y).$$

Therefore, by definition, $\mu(A^*) = \bar{\mu}(A^*)$. Similarly, $\mu(B^*) = \bar{\mu}(B^*)$.

Suppose there are n nodes in $B^* \setminus A^*$. We next show that these nodes can be arranged into a sequence $v_1, v_2, \ldots, v_n$ such that

$$\begin{cases}\phi(X_j) \leq \phi(X_{j+1}) & \text{if} \quad j < |S^g|,\\ \phi(X_j) \geq \phi(X_{j+1}) & \text{if} \quad j \geq |S^g|.\end{cases}$$

where $|S^g|$ is the size of the seed set returned by the greedy algorithm, $X_0 = A^*$, $X_{|S^g|} = S^g$, $X_n = B^*$ and $X_{j+1} = X_j \cup \{v_{j+1}\}$ for any $0 \leq j < n$.

Specifically, for the simple greedy algorithm, let $\{v_1, v_2, \ldots, v_{|S^g|}\}$ be the sequence of the seeds selected and $\{v_{|S^g|+1}, v_{|S^g|+2}, \ldots, v_n\}$ be a random sequence of the unselected nodes in $B^* \setminus A^*$. Then, for any $0 \leq j < |S^g|$, we have $\phi(X_j) \leq \phi(X_{j+1})$ since the simple greedy algorithm selects a new seed with non-negative marginal profit gain in each iteration. On the other hand, for any $|S^g| \leq j < n$, we know that $\phi(X_{|S^g|}) \geq \phi(X_{|S^g|} \cup \{v_{j+1}\})$. Due to the submodularity of $\phi(\cdot)$, we have $\phi(X_{j+1}) - \phi(X_j) \leq \phi(X_{|S^g|} \cup \{v_{j+1}\}) - \phi(X_{|S^g|}) \leq 0$ since $X_{|S^g|} \subseteq X_j$.

For the double greedy algorithms, let $\{v_1, v_2, \ldots, v_{|S^g|}\}$ be the sequence of the nodes added to S and $\{v_{|S^g|+1}, v_{|S^g|+2}, \ldots, v_n\}$ be the reverse sequence of the nodes removed from T. From Lemma II.1 in [5], we know that $r^+ + r^- \geq 0$ in each iteration of Algorithm 2. Thus, the nodes with negative r^+ must have positive r^-. Then, according to the decision condition of Algorithm 2 (line 5), these nodes must be removed from T by both the deterministic and randomized double greedy algorithms. This implies that all the nodes added to S must have non-negative r^+. As a result, for any $0 \leq j < |S^g|$, we have $\phi(v_{j+1}|X_j) = \phi(X_{j+1}) - \phi(X_j) \geq 0$ at the time when node v_{j+1} is added to S. Similarly, the nodes with negative r^- must have positive r^+. Then, these nodes must be added to S by the double greedy algorithms. This implies that all the nodes removed from T must have non-negative r^-. Consequently, for any $|S^g| \leq j < n$, we have $-\phi(v_{j+1} \mid X_j) = \phi(X_j) - \phi(X_{j+1}) \geq 0$ at the time when node v_{j+1} is removed from T.

Then, according to Proposition 8, we have $\mu(S^g) = \mu(X_{|S^g|}) \leq \mu(X_{|S^g|+1}) \leq \cdots \leq \mu(B^*) = \bar{\mu}(B^*)$ and $\bar{\mu}(S^g) = \bar{\mu}(X_{|S^g|}) \leq \bar{\mu}(X_{|S^g|-1}) \leq \cdots \leq \bar{\mu}(A^*) = \mu(A^*)$. Note that $X_0 (= A^*), X_1, \ldots, X_{|S^g|-1}$, and $X_{|S^g|+1}, X_{|S^g|+2}, \ldots, X_n (= B^*)$ are actually all the intermediate seed sets explored by the greedy algorithm. Hence, the theorem is proven. □

Theorem 4 suggests that the seed sets returned by the simple greedy and double greedy algorithms can provide tighter upper bounds than any other intermediate seed set constructed during the greedy procedure. Thus, to obtain tighter upper bounds on the optimal profit, we can directly derive upper bounds based on the seed sets returned by the greedy algorithms. We refer to these two upper bounds as $\mu_2 \triangleq \mu(S^g)$ and $\mu_3 \triangleq \bar{\mu}(S^g)$. We shall study the effectiveness of these bounds in the experimental evaluation. On the other hand, if we do not apply Theorem 4, we can still get these upper bounds by comparing those obtained from all the intermediate seed sets constructed during the greedy procedure and finding the smallest one among them. However, it would take much longer time (which increases the time complexity by a multiplicative factor of $O(|B^* \setminus A^*|)$ for computing the upper bounds) to get the same result than using Theorem 4.

TABLE 2
Statistics of OSN Datasets

Dataset	#Nodes ($\lvert V\rvert$)	#Edges ($\lvert E\rvert$)	Type	Avg. degree
Facebook	4 K	88 K	Undirected	43.7
Wiki-Vote	7 K	104 K	Directed	29.1
Google+	108 K	14 M	Directed	254.1
LiveJournal	5 M	69 M	Directed	28.5

6 Evaluation

6.1 Experimental Setup

Datasets. We use several real OSN datasets in our experiments [21]. Table 2 shows the statistics of these datasets.

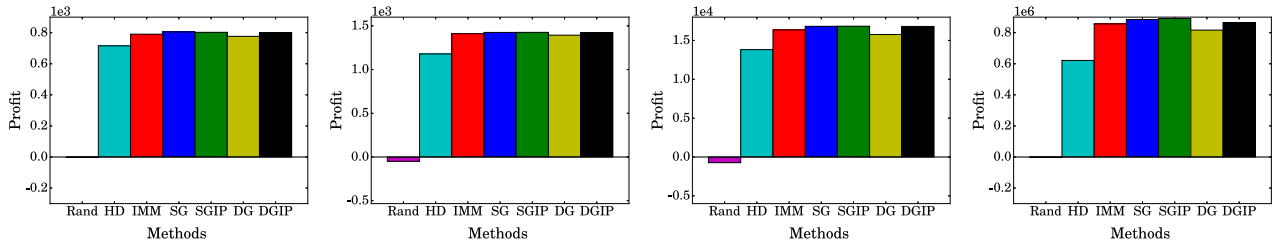


Fig. 3. Profits produced by different algorithms under uniform cost setting, uniform benefit setting, and IC model.

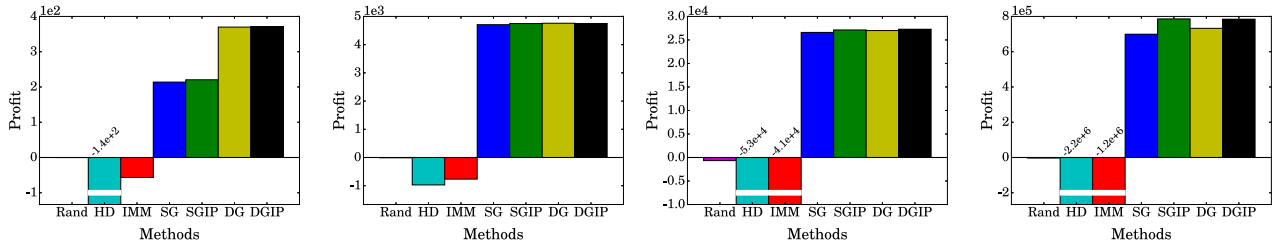


Fig. 4. Profits produced by different algorithms under degree-proportional cost setting, uniform benefit setting, and IC model.

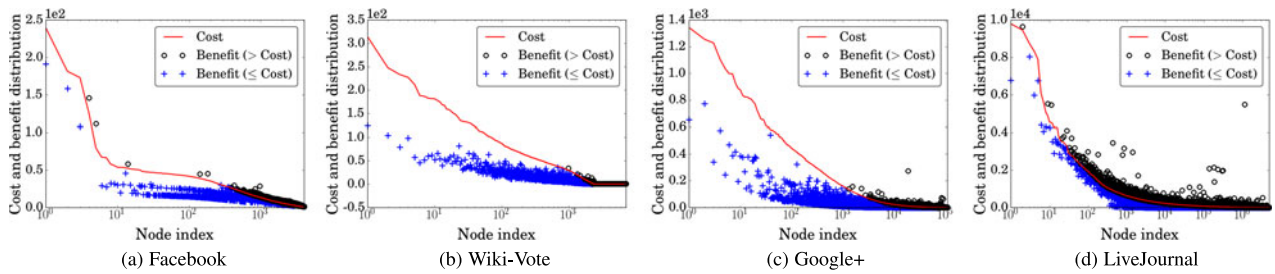


Fig. 5. Cost and benefit distributions by a single seed under degree-proportional cost setting, uniform benefit setting, and IC model.

Algorithms. The performance comparison includes the following algorithms.

- *Random* (Rand): It randomly selects a given number of k nodes. We run the algorithm 10 times and take their average as the expected profit. To explore different seed set sizes, we iterate through $k = \frac{|V|}{2^i}$ for $i = 0, 1, , 10$ (where $|V|$ is the number of nodes in the social graph) 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 $k = \frac{|V|}{2^i}$ for $i = 0, 1, \ldots, 10$.
- *IMM/BCT*: IMM [36] is a state-of-the-art sampling-based method that can provide theoretical guarantees for finding the top-k influential nodes for influence maximization, where activating every node offers the same benefit. BCT [27] is proposed to handle the scenario that the nodes have weighted benefits if activated. Thus, in our experiments, IMM is adopted in the uniform benefit setting, whereas BCT is adopted in the weighted benefit setting. We set the algorithm parameters $\epsilon = 0.5$ and $l = 1$ according to the default setting in [36] for both IMM and BCT. We again choose the k value producing the largest profit among $k = \frac{|V|}{2^i}$ for $i = 0, 1, , 10$.
- *Simple Greedy* (SG): We adopt the *Reverse Influence Sampling* (RIS) method used in IMM/BCT 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/BCT method among the above 11 cases of different k values. We also adopt the CELF technique [20] in the implementation of the simple greedy algorithm to enhance its efficiency.
- *Simple Greedy with Iterative Pruning* (SGIP): It runs simple greedy after conducting the iterative pruning as described in Section 4.3.
- *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 conducting the iterative pruning.

Parameter Settings. We use both the Independent Cascade (IC) and Linear Threshold (LT) propagation models. The propagation probability $p_{u,v}$ of each edge $\langle u, v\rangle$ is set to the reciprocal of v's in-degree, i.e., $p_{u,v} = 1/|I_v|$, as widely adopted by other studies [6], [16], [27], [28], [36], [37]. We test different benefit and cost settings. In the uniform benefit setting, every node has a unit benefit if activated. In the weighted benefit setting, we assign each node v with a benefit value randomly generated from a normal distribution with a mean 3.0 and a standard deviation 1.0, i.e.,

TABLE 3
Running Times of Different Algorithms (Seconds)

(a)Uniform Cost Setting					
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
(b) Degree-Proportional Cost Setting					
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

TABLE 4
Impact of Iterative Pruning

(a) Uniform Cost Setting				
Dataset	$\lvert A^*\rvert$	$\lvert B^*\rvert$	$\lvert B^* \setminus A^*\rvert$	$\phi(A^*) + \phi(B^*)$
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
(b) Degree-Proportional Cost Setting				
Dataset	$\lvert A^*\rvert$	$\lvert B^*\rvert$	$\lvert B^* \setminus A^*\rvert$	$\phi(A^*) + \phi(B^*)$
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

TABLE 5
Upper Bounds (Normalized by the Profit of DGIP)

(a)Uniform Cost Setting						
Dataset	μ_1^{DG}	μ_2^{DG}	μ_3^{DG}	μ_1^{DGIP}	μ_2^{DGIP}	μ_3^{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
(b) Degree-Proportional Cost Setting						
Dataset	μ_1^{DG}	μ_2^{DG}	μ_3^{DG}	μ_1^{DGIP}	μ_2^{DGIP}	μ_3^{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

$b(v) \sim N(3.0, 1.0)$. To avoid negative benefit values, we set $b(v) = 0$ if the randomly generated value is negative.[2] In the uniform cost setting, all nodes have the same costs for seed selection. In the degree-proportional cost setting, the cost of each node is set proportional to its out-degree to emulate that popular users need more incentives to participate. The ratio between the total seed selection cost of all nodes and total benefit of all nodes is controlled by a scale factor λ, i.e., $\sum_{v \in V} c(v) = \lambda \cdot \sum_{v \in V} b(v)$. 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.[3] 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.

6.2 Profits Produced by Different Algorithms

Figs. 3 and 4 show the profits produced by different algorithms under the IC model, uniform benefit setting and different cost settings. Comparing the seed selection algorithms, our greedy algorithms are more effective in optimizing the profit than the three baseline algorithms (Rand, HD and IMM). The iterative pruning technique can further improve the greedy algorithms (by up to 13 percent).

Under the uniform cost setting, our simple greedy algorithm degenerates to the IMM algorithm that iterates through all possible seed set sizes. Thus, as seen from Fig. 3, the greedy algorithms and IMM produce similar profits, and they both outperform the high degree and random algorithms. The random algorithm, in particular, generates near-zero profit and is difficult to benefit from viral marketing.

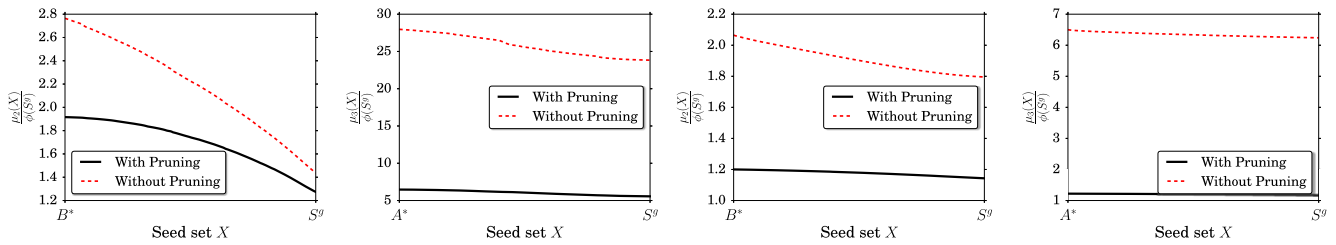
Under the degree-proportional cost setting, as seen from Fig. 4, the high degree and IMM algorithms perform even worse than the random algorithm with very negative profits produced. This is because under such a setting, the high-degree nodes have large costs. To further explore, we plot in Fig. 5 the cost distribution over all the nodes and the influence spread of each node when it is chosen as the only seed (which indicates the maximum possible influence contributed by each node in any seed set according to the submodularity). Here, the nodes are indexed in decreasing order of their out-degrees, i.e., node #1 has the highest out-degree, node #2 has the second highest out-degree and so on. It can be seen from Fig. 5 that most high-degree nodes are not profitable, i.e., their costs are larger than their influence spreads. In contrast, many low-degree nodes are profitable due to their lower costs.[4] Thus, the best seed set to maximize profit should include mostly low-degree nodes. Therefore, influence maximization algorithms have poor performance.

We can also see that the double greedy algorithms perform considerably better than the simple greedy algorithm on the Facebook dataset in Fig. 4a. This can again be explained with the cost and influence distributions shown in Fig. 5. As seen from Fig. 5a, there are a few high-degree nodes with influence larger than cost. In the simple greedy algorithm, these profitable high-degree nodes are selected first since their profits (influence spread less cost) are much higher than those of low-degree nodes. This, in consequence, would prevent many low-degree nodes from being further selected because their influence spreads largely overlap with the profitable high-degree nodes. On the other hand, the double greedy algorithm does not select the profitable high-degree nodes. Since the influence spreads of the profitable high-degree nodes overlap with the low-degree

2. The number of negative values is very small since about 99.7 percent of values drawn from a normal distribution are within three times of the standard deviation away from the mean, i.e., $\Pr[0 \leq b(v) \leq 6] \approx 99.7\%$.

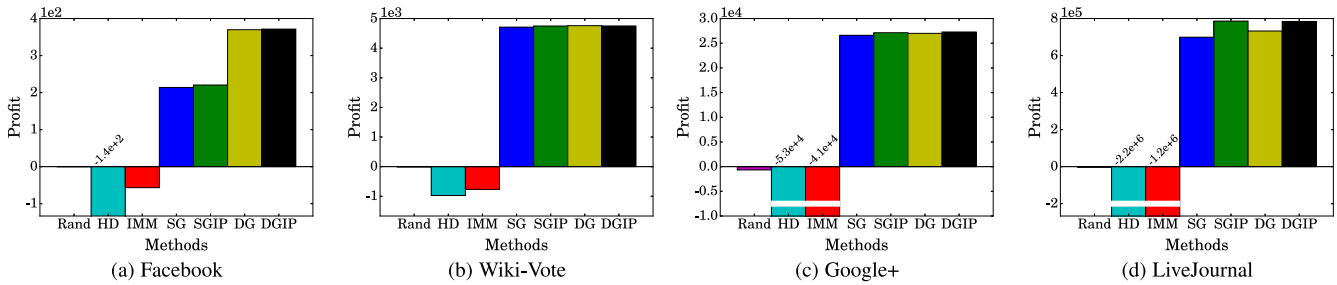
3. We have tested other values of λ and observed similar performance trends. Only the results for $\lambda = 10$ are presented due to space limitations.

4. Note that under the uniform benefit setting, the benefit is equivalent to the influence spread.



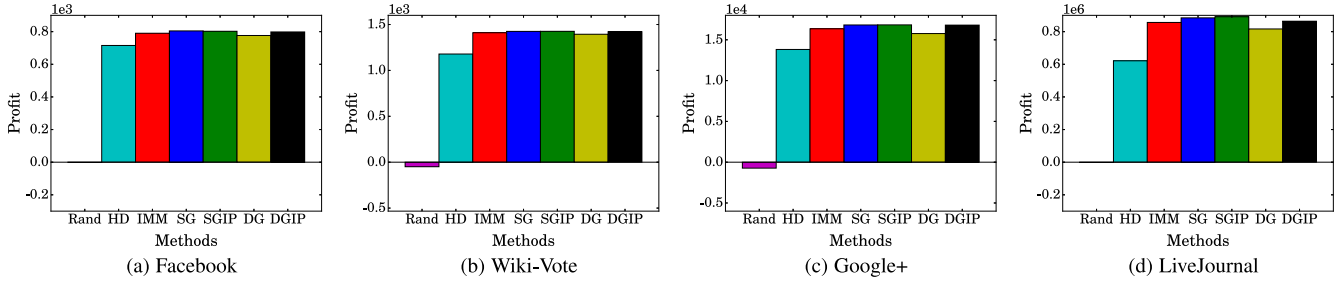
(a) μ_2 under Uniform Cost Setting (b) μ_3 under Uniform Cost Setting (c) μ_2 under Degree-Proportional Cost Setting (d) μ_3 under Degree-Proportional Cost Setting

Fig. 6. Upper bounds (normalized by the profit of DGIP) under weighted benefit setting and IC model on LiveJournal dataset.



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

Fig. 7. Profits produced by different algorithms under degree-proportional cost setting, weighted benefit setting, and IC model.



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

Fig. 8. Profits produced by different algorithms under degree-proportional cost setting, weighted benefit setting, and LT model.

nodes, when these high-degree nodes are removed from a nearly-full seed set (which includes almost all the nodes), there is not much loss in the total influence spread. Thus, the marginal profit gains r^- for these high-degree nodes to quit from the nearly-full seed set are basically their seed selection costs (line 4 of Algorithm 2). Fig. 5a shows that such costs are higher than the marginal profit gains r^+ generated by adding these high-degree nodes into an almost-empty seed set (line 3 of Algorithm 2). Thus, according to the double greedy algorithm (lines 5–8 of Algorithm 2), these high-degree nodes would not be selected. The nodes selected by the double greedy algorithm are mostly low-degree nodes which are able to generate similar total influence spread as the profitable high-degree nodes and have lower total cost than the latter. Therefore, the double greedy algorithm produces remarkably higher profit than the simple greedy algorithm. This phenomenon does not occur in the Wiki-Vote and Google+ datasets as all the high-degree nodes are not profitable (Figs. 5b and 5c). For the LiveJournal dataset, there are also a few profitable high-degree nodes. Meanwhile, there are quite many low-degree nodes offering even higher profits than these high-degree nodes. As a result, the profitable high-degree nodes are not selected by the simple greedy algorithm and the above phenomenon does not occur. These observations show that the double greedy algorithm is more robust than the simple greedy algorithm.

6.3 Running Time

Table 3 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 3.2 GHz CPU and 16 GB 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 3. It can be seen that the IMM algorithm runs significantly slower than our greedy algorithms. This is because IMM 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 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. The running times of our greedy algorithms are less than 600 seconds even for the large LiveJournal dataset with millions of nodes. This demonstrates the efficiency of our algorithms.

6.4 Iterative Pruning Technique

Table 4 summarizes the impact of the iterative pruning technique proposed in Section 4.3. As can be seen, pruning substantially reduces the number of nodes that need to be considered for seed selection (by at least 1 order of magnitude in most cases). In addition, with a scale factor $\lambda = 10$, the profit of selecting all the nodes as seeds is

$\phi(V) = |V| - 10 \cdot |V| < 0$. Thus, running the double greedy algorithm with the entire node set would not offer any approximation guarantee. In contrast, as shown in Table 4, $\phi(A^*) + \phi(B^*) > 0$ holds for most of the cases tested. In these cases, by Corollary 1, the pruning technique enables strong theoretical guarantees on the seed sets constructed by double greedy algorithms. In particular, for the degree-proportional cost setting on Wiki-Vote, we have $A^* = B^*$ 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.

6.5 Upper Bounds

Table 5 shows the upper bounds derived based on 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 $\mu_2 = \mu(S^g)$ 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 5). For the cases where $\phi(A^*) + \phi(B^*) < 0$ (see Table 4), μ_1^{DGIP} is far above 3 times the profit returned by the algorithm. In these cases, μ_2^{DGIP} certifies approximation guarantees from 46 to 78 percent for the seed set returned by the DGIP algorithm. For other cases where $\phi(A^*) + \phi(B^*) \geq 0$, all the bounds are rather close to the profit obtained by the DGIP algorithm, and μ_2^{DGIP} certifies at least 76 percent approximation guarantee for the seed set returned by the algorithm. These observations imply that the DGIP algorithm usually performs quite close to the optimal solution.

Fig. 6 shows the evolution of upper bounds derived from the intermediate seed sets explored by the DGIP algorithm. We show both the bounds derived with pruning (using A^* and B^*) and without pruning (using $\emptyset$ and V as explained in the remark of Section 5.2). It can be seen that the upper bounds are progressively tightened as the algorithm executes. The final seed set S^g returned by DGIP provides the tightest bound among all the seed sets examined. This confirms the theoretical results of Theorem 4. It can also be seen that the bounds derived with pruning are much tighter than those without pruning, which verifies the remark in Section 5.2.

6.6 Weighted Benefit Setting

Fig. 7 shows the profits produced by different algorithms under the IC model, degree-proportional cost setting, and weighted benefit setting. The results are similar to those under the uniform benefit setting (Fig. 4). Our greedy algorithms considerably outperform the three baseline algorithms (Rand, HD and BCT). The iterative pruning technique can further improve the greedy algorithms (Figs. 7c and 7d).

6.7 LT Propagation Model

As discussed, our solutions and analysis can be applied to a variety of influence propagation models. In this set of experiments, we further evaluate the algorithms with the LT propagation model. Fig. 8 shows the profits produced by different algorithms under the LT model, degree-proportional cost setting, and weighted benefit setting. In general, the results are quite similar to those under the IC model. Our greedy algorithms perform much better than the baseline algorithms, and the double greedy algorithm is more robust than the simple greedy algorithm.

7 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 differentiates the profit maximization problem from the traditional influence maximization problem. We have presented simple greedy and double greedy algorithms for seed selection, and proposed several new techniques to enhance/benchmark their performance and expand the applicability of their approximation guarantees. Experimental results with real OSN datasets show that: (1) our greedy algorithms substantially outperform several baseline algorithms; (2) our upper bounds on the maximum achievable profit offer much tighter guarantees on the quality of the solutions constructed by various algorithms; (3) our iterative pruning technique can substantially tighten the upper bounds and further improve the greedy algorithms.

Acknowledgments

This research is supported by the National Research Foundation, Prime Minister's Office, Singapore under its IDM Futures Funding Initiative, and by the Singapore Ministry of Education Academic Research Fund Tier 1 under Grant 2017-T1-002-024 and Tier 2 under Grant MOE2015-T2-2-114. A preliminary report of this work has been presented at IEEE ICNP 2016 conference [34].

References

[1] 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*, vol. 8, no. 7, pp. 814–825, 2015.

[2] E. Balkanski, A. Rubinstein, and Y. Singer, "The limitations of optimization from samples," in *Proc. 49th Annu. ACM SIGACT Symp. Theory Comput.*, 2017, pp. 1016–1027.

[3] N. Barbieri, F. Bonchi, and G. Manco, "Topic-aware social influence propagation models," in *Proc. IEEE 12th Int. Conf. Data Mining*, 2012, pp. 81–90.

[4] C. Borgs, M. Brautbar, J. Chayes, and B. Lucier, "Maximizing social influence in nearly optimal time," in *Proc. 25th Annu. ACM-SIAM Symp. Discrete Algorithms*, 2014, pp. 946–957.

[5] N. Buchbinder, M. Feldman, J. S. Naor, and R. Schwartz, "A tight linear time (1/2)-approximation for unconstrained submodular maximization," in *Proc. IEEE 53rd Annu. Symp. Found. Comput. Sci.*, 2012, pp. 649–658.

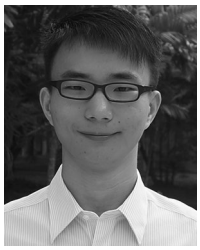
[6] W. Chen, C. Wang, and Y. Wang, "Scalable influence maximization for prevalent viral marketing in large-scale social networks," in *Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2010, pp. 1029–1038.

[7] W. Chen, Y. Wang, and S. Yang, "Efficient influence maximization in social networks," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2009, pp. 199–208.

[8] G. Cordasco, L. Gargano, and A. A. Rescigno, "Influence propagation over large scale social networks," in *Proc. IEEE/ACM Int. Conf. Advances Social Netw. Anal. Mining*, 2015, pp. 1531–1538.

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

[10] U. Feige, V. S. Mirrokni, and J. Vondrák, "Maximizing non-monotone submodular functions," in *Proc. 48th Annu. IEEE Symp. Found. Comput. Sci.*, 2007, pp. 461–471.
[11] A. Goyal, F. Bonchi, and L. V. Lakshmanan, "Learning influence probabilities in social networks," in *Proc. 3rd ACM Int. Conf. Web Search Data Mining*, 2010, pp. 241–250.
[12] A. Goyal, F. Bonchi, L. Lakshmanan, and S. Venkatasubramanian, "On minimizing budget and time in influence propagation over social networks," *Social Netw. Anal. Mining*, vol. 3, no. 2, pp. 179–192, 2013.
[13] J. Hartline, V. Mirrokni, and M. Sundararajan, "Optimal marketing strategies over social networks," in *Proc. 17th Int. Conf. World Wide Web*, 2008, pp. 189–198.
[14] R. Iyer and J. Bilmes, "Submodular optimization with submodular cover and submodular knapsack constraints," in *Proc. 26th Int. Conf. Neural Inf. Process. Syst.*, 2013, pp. 2436–2444.
[15] R. Iyer, S. Jegelka, and J. Bilmes, "Fast semidifferential-based submodular function optimization," in *Proc. 30th Int. Conf. Mach. Learning*, 2013, pp. 855–863.
[16] K. Jung, W. Heo, and W. Chen, "IRIE: Scalable and robust influence maximization in social networks," in *Proc. IEEE 12th Int. Conf. Data Mining*, 2012, pp. 918–923.
[17] D. Kempe, J. Kleinberg, and E. Tardos, "Maximizing the spread of influence through a social network," in *Proc. 9th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2003, pp. 137–146.
[18] K. Kutzkov, A. Bifet, F. Bonchi, and A. Gionis, "STRIP: Stream learning of influence probabilities," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2013, pp. 275–283.
[19] J. Lee, V. S. Mirrokni, V. Nagarajan, and M. Sviridenko, "Maximizing nonmonotone submodular functions under matroid or knapsack constraints," *SIAM J. Discrete Math.*, vol. 23, no. 4, pp. 2053–2078, 2010.
[20] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance, "Cost-effective outbreak detection in networks," in *Proc. 13th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2007, pp. 420–429.
[21] J. Leskovec and A. Krevl, "SNAP Datasets: Stanford large network dataset collection," 2014. [Online]. Available: http://snap.stanford.edu/data
[22] Y. Li, B. Q. Zhao, and J. C. S. Lui, "On modeling product advertisement in large-scale online social networks," *IEEE/ACM Trans. Netw.*, vol. 20, no. 5, pp. 1412–1425, 2012.
[23] C. Long and R. C.-W. Wong, "Minimizing seed set for viral marketing," in *Proc. IEEE 11th Int. Conf. Data Mining*, 2011, pp. 427–436.
[24] W. Lu and L. V. Lakshmanan, "Profit maximization over social networks," in *Proc. IEEE 12th Int. Conf. Data Mining*, 2012, pp. 479–488.
[25] H. Narasimhan, D. C. Parkes, and Y. Singer, "Learnability of influence in networks," in *Proc. 28th Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 3186–3194.
[26] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions-I," *Math. Programm.*, vol. 14, no. 1, pp. 265–294, 1978.
[27] H. T. Nguyen, T. N. Dinh, and M. T. Thai, "Cost-aware targeted viral marketing in billion-scale networks," in *Proc. IEEE INFOCOM*, 2016, pp. 1–9.
[28] H. T. Nguyen, M. T. Thai, and T. N. Dinh, "Stop-and-stare: Optimal sampling algorithms for viral marketing in billion-scale networks," in *Proc. Int. Conf. Manage. Data*, 2016, pp. 695–710.
[29] N. Ohsaka, T. Akiba, Y. Yoshida, and K. Kawarabayashi, "Fast and accurate influence maximization on large networks with pruned monte-carlo simulations," in *Proc. 28th AAAI Conf. Artif. Intell.*, 2014, pp. 138–144.
[30] M. G. Rodriguez, D. Balduzzi, and B. Schölkopf, "Uncovering the temporal dynamics of diffusion networks," in *Proc. Int. Conf. Mach. Learning*, 2011, pp. 561–568.
[31] D. Sheldon, et al., "Maximizing the spread of cascades using network design," in *Proc. 26th Conf. Uncertainty Artif. Intell.*, 2010, pp. 517–526.
[32] 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 Distrib. Syst.*, vol. 26, no. 5, pp. 1379–1392, May 2015.
[33] J. Tang, X. Tang, X. Xiao, and J. Yuan, "Online processing algorithms for influence maximization," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2018, pp. 1–15.
[34] J. Tang, X. Tang, and J. Yuan, "Profit maximization for viral marketing in online social networks," in *Proc. IEEE 24th Int. Conf. Netw. Protocols*, 2016, pp. 1–10.
[35] J. Tang, X. Tang, and J. Yuan, "Influence maximization meets efficiency and effectiveness: A hop-based approach," in *Proc. IEEE/ACM Int. Conf. Advances Social Netw. Anal. Mining*, 2017, pp. 64–71.
[36] Y. Tang, Y. Shi, and X. Xiao, "Influence maximization in near-linear time: A martingale approach," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2015, pp. 1539–1554.
[37] Y. Tang, X. Xiao, and Y. Shi, "Influence maximization: Near-optimal time complexity meets practical efficiency," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2014, pp. 75–86.
[38] 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. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2014, pp. 1306–1315.
[39] 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 13th Int. Conf. Data Mining*, 2013, pp. 907–916.
[40] Y. Zhu, Z. Lu, Y. Bi, W. Wu, Y. Jiang, and D. Li, "Influence and profit: Two sides of the coin," in *Proc. IEEE 13th Int. Conf. Data Mining*, 2013, pp. 1301–1306.



Jing Tang (S'16) received the BEng degree from the University of Science and Technology of China (USTC) in 2012 and the PhD degree from Nanyang Technological University (NTU) in 2017, both majored in computer science. He is currently a research fellow at Nanyang Technological University (NTU), Singapore. His research interests include big data management and analytics, online social networks, distributed systems, and network economics. He received the Best Paper Award from the IEEE International Conference on Network Protocols 2014 (ICNP'14). He is a student member of the IEEE.

Xueyan Tang (M'04–SM'09) received the BEng degree in computer science and engineering from Shanghai Jiao Tong University, in 1998, and the PhD degree in computer science from the Hong Kong University of Science and Technology, in 2003. He is currently an associate professor in the School of Computer Science and Engineering, Nanyang Technological University, Singapore. His research interests include distributed systems, cloud computing, mobile and pervasive computing, and wireless sensor networks. He has served as an associate editor of the *IEEE Transactions on Parallel and Distributed Systems*, and a program co-chair of IEEE ICPADS 2012 and CloudCom 2014. He is a senior member of the IEEE.

Junsong Yuan (M'08–SM'14) received the PhD from Northwestern University and the MEng degree from the National University of Singapore. Before that, he graduated from the Special Class for the Gifted Young of Huazhong University of Science and Technology (HUST), Wuhan, China, in 2002. He is currently an associate professor in the School of Electrical and Electronics Engineering (EEE), Nanyang Technological University (NTU). His research interests include computer vision, video analytics, gesture and action analysis, and large-scale visual search and mining. He received best paper award from the International Conference on Advanced Robotics (ICAR'17), 2016 Best Paper Award from IEEE Transactions on Multimedia, Doctoral Spotlight Award from IEEE Conference on Computer Vision and Pattern Recognition (CVPR'09), Nanyang Assistant Professorship from NTU, and Outstanding EECS Ph.D. Thesis award from Northwestern University. He is currently an associate editor of the *IEEE Transactions on Image Processing (T-IP)*, the *IEEE Transactions on Circuits and Systems for Video Technology (T-CSVT)*, the *Journal of Visual Communications and Image Representations (JVCI)*, and *The Visual Computer* Journal (TVC), and served as guest editor of the *International Journal of Computer Vision (IJCV)*. He is/was program co-chair of ICME'18 and VCIP'15, and area chair of CVPR'17, ICIP'17, ICPR'16, ICME'15'14, ACCV'14, and WACV'14. He is a senior member of the IEEE.