Zhang BL, Qian ZZ, Li WZ *et al.* Budget allocation for maximizing viral advertising in social networks. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 31(4): 759–775 July 2016. DOI 10.1007/s11390-016-1661-3

Budget Allocation for Maximizing Viral Advertising in Social Networks

Bo-Lei Zhang [1], *Student Member, CCF*, Zhu-Zhong Qian [1,*], *Member, CCF, ACM, IEEE*
Wen-Zhong Li [1], *Member, CCF, ACM, IEEE*, Bin Tang [1], *Member, CCF, ACM, IEEE*
Sang-Lu Lu [1], *Senior Member, CCF, Member, ACM, IEEE*, and
Xiaoming Fu [2], *Senior Member, IEEE, Member, ACM*

[1] *State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China*
[2] *Institute of Computer Science, University of Göttingen, Göttingen 37077, Germany*

E-mail: zhangbolei@dislab.nju.edu.cn; {qzz, lwz, tb, sanglu}@nju.edu.cn; fu@cs.uni-goettingen.de

Received February 29, 2016; revised May 12, 2016.

Abstract Viral advertising in social networks has arisen as one of the most promising ways to increase brand awareness and product sales. By distributing a limited budget, we can incentivize a set of users as initial adopters so that the advertising can start from the initial adopters and spread via social links to become viral. Despite extensive researches in how to target the most influential users, a key issue is often neglected: how to incentivize the initial adopters. In the problem of influence maximization, the assumption is that each user has a fixed cost for being initial adopters, while in practice, user decisions for accepting the budget to be initial adopters are often probabilistic rather than deterministic. In this paper, we study optimal budget allocation in social networks to maximize the spread of viral advertising. In particular, a concave probability model is introduced to characterize each user's utility for being an initial adopter. Under this model, we show that it is NP-hard to find an optimal budget allocation for maximizing the spread of viral advertising. We then present a novel discrete greedy algorithm with near optimal performance, and further propose scaling-up techniques to improve the time-efficiency of our algorithm. Extensive experiments on real-world social graphs are implemented to validate the effectiveness of our algorithm in practice. The results show that our algorithm can outperform other intuitive heuristics significantly in almost all cases.

Keywords social network, influence maximization, information diffusion, submodular optimization

1 Introduction

The last decade has witnessed the emergence and proliferation of online social networks, such as Facebook, Twitter and Youtube. People have been actively engaged in the networks and generating contents at an ever-increasing rate. The online social networks serving as new platforms are providing great opportunities for the widespread information among individuals. Viral advertising, which utilizes information diffusion in social networks for the promotions of new products, ideas and innovations, is attracting enormous attentions from advertisers and providers. Compared with TVs, newspapers and radios which broadcast information, viral advertising in social networks has the "word-of-mouth" effect, and is considered to be more trustworthy. Moreover, the viral advertising can spread across multiple links and trigger large cascades of adoption. Such characteristics have made the viral advertising more and more popular in business campaigns.

Suppose we are given a limited budget. To maximize the spread of viral advertising, we would first

Regular Paper

Special Section on Data Management and Data Mining 2016

This work was partially supported by the National Natural Science Foundation of China under Grant Nos. 61373128, 61321491, 61472181, 91218302, the Natural Science Foundation of Jiangsu Province of China under Grant No. BK20151392, Jiangsu Key Technique Project (Industry) under Grant No. BE2013116, EU FP7 IRSES MobileCloud Project under Grant No. 612212, the Program B for Outstanding Ph.D. Candidate of Nanjing University, and the Collaborative Innovation Center of Novel Software Technology and Industrialization of Jiangsu Province of China.

*Corresponding Author

distribute the budget as a vector of incentives among the users, expecting that a seed set of them would accept the allocated budget to be initial adopters. The initial adopters will then spread the advertising to their friends as an information cascade. One of the fundamental problems adopting this idea is influence maximization[1], which aims to select the most influential set of users as initial adopters so that users involved in the cascade can be maximized. Despite that much effort has been devoted to optimizing the selection of initial adopters[2-5], a key issue is often neglected: how should we incentivize the initial adopters?

In influence maximization, the underlying assumption is that each user has a deterministic cost for being initial, i.e., a user can be incentivized as long as the allocated budget exceeds the deterministic cost. However, in practice, user decisions for being initial adopters are often probabilistic rather than deterministic. With different amount of allocated budget, the users may have different probabilities to be initial adopters. For the above reason, it is necessary to understand how users make decisions for adopting the allocated budget. Here, we introduce utility model to characterize users' satisfaction for being initial adopters with the incentives. Utility has been widely adopted in economics and game theory as a measure of preferences over goods and services, and can often be described as non-linear concave functions[6-7].

Under this model, the process of viral advertising can be summarized in two phases. In the first phase, a set of users may accept the allocated budget to be initial adopters according to the utility model. In the second phase, the advertising will start from the initial adopters and spread in the social network as a piece of information. Fig.1 shows an illustrative example of the two phases. The number of active users after the cascade is called the spread of the viral advertising. To maximize the spread of viral advertising, we must determine to whom we shall allocate budget and how much should be allocated to each of them.

In this paper, we study optimal budget allocation in social networks to maximize the spread of viral advertising. Specifically, we introduce utility functions to model the user satisfaction for the allocated budget. The advertising would then spread according to the Independent Cascade (IC) model[1]. The aim is to maximize the spread of viral advertising. We first show that the problem can be reduced from existing NP-hard problems. Due to the hardness of the problem and non-concavity of the objective function, we first consider a simpler discrete setting and propose a greedy algorithm with performance guarantee. We further show that the algorithm can be proved with near optimal result even in the general continuous setting if the budget is discretized properly. Several scaling-up techniques are proposed to improve the time efficiency of the algorithm. Finally we conduct extensive evaluations for validation.

1.1 Our Results

The main contributions of this paper are summarized as follows.

• We introduce probabilistic functions to model user utility for the allocated budget. The model is practical and can characterize different users' preferences for being initial adopters in social networks. Under such model, we formalize the budget allocation problem as

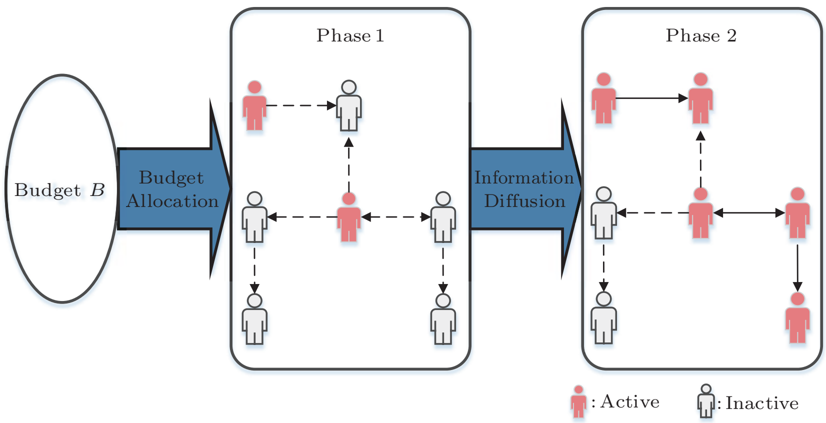


Fig.1. Two phases of the budget allocation for viral advertising. In the first phase, the red users are incentivized as initial adopters. In the second phase, the initial adopters would spread the information to other users. Note that the two phases are inherently probabilistic.

an optimization problem and show that it can be reduced from existing NP-hard problems.

• Due to the hardness, we propose a discrete greedy algorithm. We prove that in the discrete setting, the greedy algorithm achieves $1-1/e$ approximation ratio, and in the more general continuous setting, when the budget is discretized in $O(n)$ pieces, the algorithm can reach $1-1/e-o(1)$ approximation ratio, where $o(1)$ is a relatively small error.

• We improve the time efficiency with scaling-up techniques. Moreover, we design a novel algorithm for estimating the pairwise diffusion probability in IC model.

• Extensive experiments on different datasets are implemented to validate the effectiveness of our algorithm. The results show that our algorithm outperforms other intuitive algorithms significantly in almost all cases. In addition, the algorithm has relatively higher time-efficiency.

1.2 Related Work

Our work has a strong tie with the problem of influence maximization, which was first proposed by Domingos and Richardson[8-9]. Suppose we would like to market a new product to be adopted by a large fraction of the network. They posed the idea that by giving the "influential" initial adopters free samples can trigger a large cascade. Later, Kempe *et al.*[1] formulated the problem as a discrete optimization problem by modeling the influence spread as IC model or Linear Threshold (LT) model. They showed that finding the optimal solution in these models is NP-hard and proposed greedy algorithm with hill-climbing strategies to find the influential nodes. Due to the monotonicity and submodularity of the diffusion models, the algorithm can be proved to achieve constant approximation ratio. Following their work, extensive researches[2-5,10-11] have studied the algorithmic improvement of the spread of influence in social networks. In [2], the authors proposed a lazy-forward technique which can accelerate the greedy algorithm for about 700 times faster. We will also adopt this idea in our algorithm. Chen *et al.*[3,12] proved that computing the influence spread in IC and LT models is #P-hard. Despite a lot of algorithmic progress in selecting the most influential initial adopters, one key issue of how to incentivize the initial nodes has been largely overlooked.

Another thread of our work is inspired by the problem of revenue maximization which was first introduced by Hartline *et al.*[13] In order to influence many buyers to buy a product, a seller could first offer some popular buyers discounts, instead of "giving them free samples". The problem then studies marketing strategies like how large the discounts should be and in what sequence the selling should happen. Since then, a lot of following researches have studied revenue maximization in social networks[14-17]. For example, Candogan *et al.*[18] studied optimal uniform budget allocation to maximize the profit of a seller. In particular, they considered each consumer's usage level depends directly on the usage of their neighbors in the social network. In [19], the authors studied iterative pricing strategies for revenue maximization. They also considered the Bayesian setting in which there is prior knowledge of the probability distribution on the valuations of buyers. Arthur *et al.*[15] considered cascading manner through the network for revenue maximization, i.e., a user is offered the product via recommendations from his/her neighbors. In [20] and [21], the authors proposed that it is more effective if we offer incentives after influence maximization and only the users who adopted the incentives should be counted. In comparison, we offer budget before influence maximization and argue that it is also very important to incentivize the initial adopters.

In a recent work[22], Singer considered auction-based influence maximization in which each user can bid a cost for being an initial adopter. Singer[22] designed incentive compatible mechanisms to make sure that each user declares the true cost. However, the mechanism requires an extra step for each user to bid a cost which may be cumbersome to implement for the advertisers. Comparatively, we adopt a budget allocation with a more natural way for incentivizing each user. In [23], the authors proposed to incorporate rewards as incentives into influence propagation and adoption group to stimulate active users. In [4], Demaine *et al.* proposed partial incentives in social networks to influence people. In comparison, we propose more general utility and influence models and further extend the results in the discrete setting. In a recent study[24], the authors tackled the general problem of influence maximization in the continuous setting. We further assume that the utility functions are concave and show the approximation of our proposed algorithm.

1.3 Organization

The rest of this paper is organized as follows. In Section 2, we introduce the models for budget allocation and formalize the optimization problem. In Section 3, we present our main discrete greedy budget allocation and show its approximation. We further scale up the algorithm in Section 4. Section 5 presents the evaluations. Finally, we conclude our paper in Section 6.

2 System Model

Consider a social network modeled as a directed (or undirected) graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N} = \{1, 2, \ldots, n\}$ denotes the set of n users in the network, and $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$ denotes the social links between users. Given a limited budget B, an allocation is then a vector of non-negative numbers $\boldsymbol{b} = (b_1, b_2, ..., b_n)$, $\sum b_i \leqslant B$ among users, where b_i is the budget allocated to user i. In the following, we first introduce a probabilistic utility model to characterize each user's satisfaction for being initial adopters, and then introduce how to model the process of viral advertising spread in a social network. Finally, the budget allocation for viral advertising is formulated as an optimization problem.

2.1 Utility Model

When distributing the budget as incentives, we introduce probability function $F_i : \mathbb{R}^+ \rightarrow [0, 1]$ to characterize user i's satisfaction for being an initial adopter, i.e., user i will accept the allocated budget b_i with probability $F_i(b_i)$. A natural property of $F_i(x)$ is monotonically increasing, i.e., a user is more likely to be an initial adopter if i is offered more budget. Formally, we have that:

$$F_i(x_1) \leqslant F_i(x_2), \quad \text{for any } x_1 < x_2.$$

As observed in literatures[6-7], the user satisfaction usually follows the law of diminishing returns, i.e., the marginal gain of user satisfaction decreases as the budget increases. Therefore, we assume that the utility function $F_i(x)$ satisfies the property of concavity:

$$\begin{aligned} & F_i(\lambda x + (1-\lambda)y) \\ \geqslant\ & \lambda F_i(x) + (1-\lambda)F_i(y), \quad \text{for any } \lambda \in [0, 1]. \end{aligned}$$

Note that different users may have different utility functions (see Fig.2 for some of the possible examples). Currently, most existing studies assume that the utility functions are known in advance[7,24]. In practice, with the growing availability of personalized data, the utility functions could be estimated or learned from the historical datasets with advanced machine learning approaches, such as maximum likelihood[25-26].

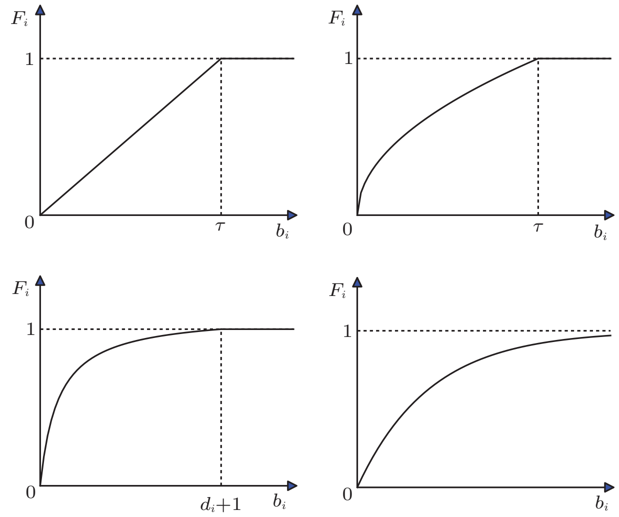


Fig.2. Examples of user utility functions.

2.2 Information Diffusion Model

After allocating the budget as incentives, a set of users would adopt the incentives to be initial adopters. The viral advertising will then start from the initial adopters to spread as an information cascade. We use the Independent Cascade (IC) model[1] for the information diffusion process, which has been widely used in the literatures[1,27-30].

IC Model. In the cascade, a node is said to be active if it adopts the information and inactive otherwise. At the beginning, only the initial adopters of S are active and they start by sharing the advertising to trigger the information cascade. Let S^t denote the set of newly activated nodes at step t, where $S^0 = S$ is the set of initial adopters. At step t, each node v, which is a neighbor of node $i \in S^t$, will be activated independently with probability $\mu_{(u,v)}$. The whole process terminates once $S^t = \emptyset$ for some t. The number of all active users after the information cascade is called the spread of the viral advertising.

Properties. Let $q(\boldsymbol{b}_{-i}, i)$ be the probability that user i can be influenced by others with budget allocation $\boldsymbol{b}_{-i}$, where $\boldsymbol{b}_{-i}$ is the allocated budget vector except b_i. The function has the following properties.

Claim 1. *For the diffusion function $q(\boldsymbol{b}_{-i}, i)$, there is:*

- *for any $j \in \mathcal{N}$, $\frac{\partial q}{\partial b_j} \geqslant 0$;*
- *for any $j, k \in \mathcal{N}$ (possibly equal), $\frac{\partial^2 q}{\partial b_j \partial b_k} \leqslant 0$.*

Proof. We use X to denote a fixed result of the graph where each edge is a coin flipped with probability μ_e. Based on the Monte Carlo method introduced in [1], the diffusion function can be formulated as the sum of active probability in all the fixed graphs:

$$q(\boldsymbol{b}_{-i}, i) = \sum_X P[X] q_X(\boldsymbol{b}_{-i}, i), \tag{1}$$

where $q_X(\boldsymbol{b}_{-i}, i)$ is the reachability probability that i is influenced in the fixed graph X via live-edges, and $P[X]$ is the probability that we get a result of X. Suppose X_i is the connected component of i. Then $q_X(\boldsymbol{b}_{-i}, i)$ can be formulated as $q_X(\boldsymbol{b}_{-i}, i) = 1 - \prod_{j \in X_i}(1 - F_j(b_j))$.

From the above formulations, we can get the following properties of the function $q(\boldsymbol{b}_{-i}, i)$. First, the diffusion function is monotone with the allocated budget $\boldsymbol{b}$:

$$\begin{aligned}\frac{\partial q(\boldsymbol{b}_{-i}, i)}{\partial b_k} &= \sum_X P[X] \frac{\partial F_k(b_k)}{\partial b_k} \\ &\quad \prod_{j \in X_i, j \neq k} (1 - F_j(b_j)) \\ &\geqslant 0,\end{aligned}$$

by the monotonicity of F. Since F is concave, there is:

$$\begin{aligned}\frac{\partial^2 q(\boldsymbol{b}_{-i}, i)}{\partial b_k^2} &= \sum_X P[X] \frac{\partial^2 F_k(b_k)}{\partial b_k^2} \\ &\quad \prod_{j \in X_i, j \neq k} (1 - F_j(b_j)) \\ &\leqslant 0.\end{aligned}$$

And the second partial derivative of q with respect to k and l ($k \neq l$) is:

$$\begin{aligned}\frac{\partial^2 q(\boldsymbol{b}_{-i}, i)}{\partial b_k \partial b_l} &= -\sum_X P[X] \frac{\partial F_k(b_k)}{\partial b_k} \frac{\partial F_l(b_l)}{\partial b_l} \\ &\quad \prod_{j \in X_i, j \neq k, j \neq l} (1 - F_j(b_j)) \\ &\leqslant 0.\end{aligned}$$

□

Remark. It is worth mentioning that many of our results can be extended to other information diffusion models, such as Coverage model[22] and Linear Threshold (LT) model[1], as long as the models have the same properties as Claim 1. Some of the results will be introduced in Subsection 3.4.

2.3 Problem Formulation

From the perspective of advertisers, one natural objective is to maximize the number of users that adopt the advertising by initializing some users as early adopters. The problem of finding the optimal budget allocation can be stated as follows.

Given a limited budget B, our goal is to find an optimal budget allocation $\boldsymbol{b} = (b_1, b_2, \ldots, b_n)$, $\sum_{i=1}^n b_i \leqslant B$, such that the expected number of active users after the information diffusion process, denoted by $f(\boldsymbol{b})$, is maximized.

For a budget allocation $\boldsymbol{b}$, let $w_i(\boldsymbol{b})$ denote the expectation that user i becomes active after the information diffusion process. Clearly, there are two independent ways for a user i to get active: i may accept the allocated budget b_i with probability $F_i(b_i)$; or i could be influenced by other initial adopters with probability $q(\boldsymbol{b}_{-i}, i)$. Fig.3 shows an illustrative example. Thus, we have

$$w_i(\boldsymbol{b}) = 1 - (1 - F_i(b_i))(1 - q(\boldsymbol{b}_{-i}, i)).$$

By the linearity of expectation, it is straightforward to see that

$$f(\boldsymbol{b}) = \sum_{i=1}^n w_i(\boldsymbol{b}).$$

Therefore, the problem of finding the optimal budget allocation can be represented as the following equivalent formulation:

$$\begin{aligned}\max \quad & f(\boldsymbol{b}) = \sum_{i=1}^n w_i(\boldsymbol{b}) \\ \text{s.t.} \quad & \sum_{i=1}^n b_i \leqslant B.\end{aligned}$$

Some of the important notations are listed in Table 1.

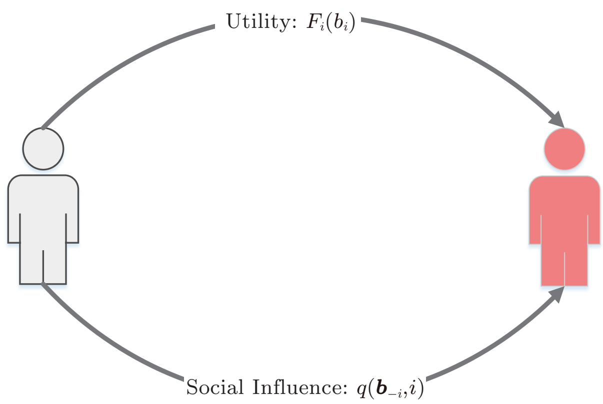


Fig.3. User states in the viral advertising. A user can convert from inactive to active in two independent ways: the utility of the allocated budget and the social influence.

Table 1. Notations

Symbol	Description
$\mathcal{G}=(\mathcal{N},\mathcal{E})$	Social network
n	Number of users or $\lvert\mathcal{N}\rvert$
m	Budget is divided into m equal pieces
d_i	Degree (indegree) of user i
B	Total budget
$\boldsymbol{b}$	Allocated budget vector
$F_i(b_i)$	Utility function of user i with respect to b_i
$q(\boldsymbol{b}_{-i}, i)$	Probability that user i is influenced by allocating $\boldsymbol{b}_{-i}$
μ_e	Influence probability on edge e
$w_i(\boldsymbol{b})$	Probability that user i will be active with $\boldsymbol{b}$
$f(\boldsymbol{b})$	Expected number of active users after information cascade by allocating $\boldsymbol{b}$

3 Optimal Budget Allocation for Viral Advertising

In this section, we first prove the NP-hardness of this problem. As the general form of the problem is intractable, we begin with a simpler discrete setting and present an approximate algorithm to the optimal. The performance guarantee of the algorithm in the general continuous setting will be analyzed in Subsection 3.3. In the last part, we will show that our algorithm can be easily extended to some other widely used submodular information diffusion models.

3.1 NP-Hardness

We start by showing the hardness of the above optimization problem. The hardness can be proved by a reduction from an existing NP-hard problem.

Theorem 1. *Identifying the optimal budget allocation is NP-hard.*

Proof. We first define a special instance of the budget allocation problem. Let $\mathcal{G}=(\mathcal{N},\mathcal{E})$ be an undirected graph and the budget B be a natural number: $B=k\in\mathbb{N}_{\geqslant 0}$. The utility function is defined as linear $F_i(b_i)=b_i$ for any $i\in\mathcal{N}$. Since $\frac{\partial^2 F_i}{\partial b_i^2}=0$, according to Claim 1, the second partial derivative of q (defined in (1)) with respect to b_i is $\frac{\partial^2 q}{\partial b_i^2}=0$. Thus it is easy to conclude that $\frac{\partial f}{\partial b_i}\geqslant 0$. In addition, there is $\frac{\partial^2 f}{\partial b_i^2}=0$ and $\frac{\partial^2 f}{\partial b_i \partial b_j}\leqslant 0$.

Now we show by contradiction that the optimal budget allocation must be a binary vector in this special case, i.e., $b_i\in\{0,1\}$ for every $i\in\mathcal{N}$. Assume the opposite is true: there are at least two users u and v with $b_u, b_v\in(0,1)$ in the optimal budget allocation. Without loss of generality, suppose that $\frac{\partial f}{\partial b_v}\geqslant\frac{\partial f}{\partial b_u}$. If we reallocate b_u to b_v until $b_v=1$, the inequality will still hold since $\frac{\partial f}{\partial b_v}$ is increasing and $\frac{\partial f}{\partial b_u}$ is decreasing by the result that $\frac{\partial^2 f}{\partial b_u \partial b_v}\leqslant 0$. However, this will generate a value of $f(\cdot)$ at least as high as the optimal, which violates the assumption.

Consider any instance of the influence maximization[1] problem: it is NP-hard to find k seed nodes to maximize the spread of influence. If there is an optimal binary budget allocation with budget k, we can get the seed nodes by selecting the nodes with $b_i=1$ to maximize the influence spread. Conversely, if there is a seed set S, we pay the nodes in S with $b_i=1$. Then, the budget allocation can reach the optimal result in the viral advertising. Thus, identifying the optimal budget allocation in the special case is equivalent to finding k seed nodes for influence maximization, which is NP-hard. □

3.2 DiscreteGreedy Algorithm

Due to the hardness of the problem, we first consider a simpler discrete setting for optimal budget allocation. Given a limited budget B, in a discrete setting, we divide it into m equal pieces, which is B/m for each. The problem is to allocate the m pieces of budget to n users to maximize the spread of viral advertising, where each user can be allocated multiple pieces of the budget. Here, m is a predefined value and the performance of our algorithm may vary with different values of m. We will further analyze how to choose a proper value of m in Subsection 3.3. Fig.4 presents an illustration of the discrete setting of the budget allocation problem.

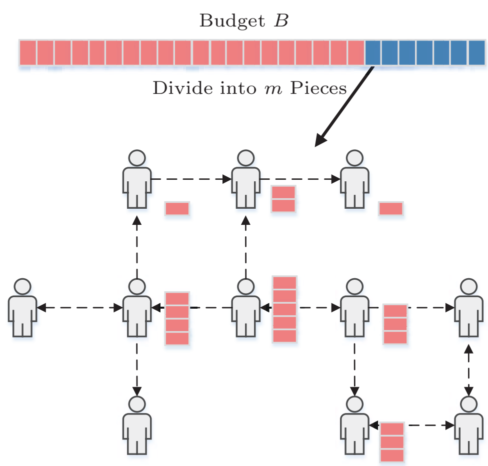


Fig.4. Illustration of the budget allocation in discrete setting. The budget is divided uniformly into m pieces and can be allocated to the users iteratively.

Algorithm. In the discrete setting, we first reduce the objective to a set function: defining a new ground set $\mathcal{A} = \mathcal{N} \times [m] = \{a_{11}, a_{12}, ..., a_{1m}, a_{21}, a_{22}, ..., a_{2m}, ..., a_{n1}, a_{n2}, ..., a_{nm}\}$, where a_{ij} means assigning the j-th piece budget to user i and $|a_{ij}| = B/m$. In set $\mathcal{A}$, the j-th piece of budget element is replicated n times as $\{a_{1j}, a_{2j}, ..., a_{nj}\}$, but only one of them can be chosen. And we need to choose a subset $A = \bigcup_{i \in \mathcal{N}} A_i$ of size m from $\mathcal{A}$ to maximize the spread of viral advertising, where A_i is the set of budget elements allocated to user i ($|A_i| = b_i$). In this discrete setting, we propose a greedy algorithm which starts from an empty set and incrementally adds budget elements for allocation. In the j-th iteration, we add a_{sj} to the user who could get the largest marginal gain $i = \arg\max_{s \in \mathcal{N}}(f(A \cup \{a_{sj}\}) - f(A))$. Slightly abusing notations, we use the budget vector $\boldsymbol{b}$ and the budget set A in a similar way.

Analysis. We show the approximation ratio of the above algorithm by viewing it as a special case of maximizing a monotone submodular function subject to a matroid① constraint. A set function $f(\cdot)$ is submodular if for any element $a \in \mathcal{A} \backslash T$, for all $U \subseteq T$, there is $f(U \cup \{a\}) - f(U) \geqslant f(T \cup \{a\}) - f(T)$. Submodularity implies that the marginal gain of choosing an element decreases as the number of chosen elements increases. First, the matroid $\mathcal{M} = (\mathcal{A}, \mathcal{I})$ can be defined as:

$$\mathcal{I} = \{A \subseteq \mathcal{A} | A \cap (\mathcal{N} \times \{j\}) \leqslant 1,\ \forall j\}.$$

Next, we show that the objective function is monotone and submodular with respect to the allocated budget.

Lemma 1. *The objective function $f(\cdot)$ in the discrete setting is monotone and submodular.*

Proof. Monotonicity. Obviously, by adding any new budget element a_{ij}, namely assigning the j-th piece of budget to user i, the probability that user i could get active will not decrease. Accordingly, the probability that user i influences other users will also not decrease, which concludes that $f(\cdot)$ is non-decreasing monotone.

Submodularity. For submodularity, let $\delta(a_{ij}|A)$ be the marginal gain by adding element a_{ij} to set A. It can be formulated as the sum of increased probability from all related users:

$$\begin{aligned}\delta(a_{ij}|A) &= f(A \cup \{a_{ij}\}) - f(A) \\ &= \delta_i(a_{ij}|A) + \sum_{k \in \mathcal{N} \backslash \{i\}} \delta_k(a_{ij}|A),\end{aligned}$$

where $\delta_k(a_{ij}|A)$ is the increased activation probability of user k by adding a_{ij} in set A. Since the class of submodular functions is closed under non-negative linear combinations, function $f(\cdot)$ can be proved to be submodular if the marginal gain of each user is non-increasing.

For user i, by adding a budget element a_{ij}, the increased probability is:

$$\begin{aligned}\delta_i(a_{ij}|A) &= w_i(A \cup \{a_{ij}\}) - w_i(A) \\ &= (F_i(A_i \cup \{a_{ij}\}) - F_i(A_i))(1 - q(A_{-i}, i)),\end{aligned}$$

where A_{-i} is the allocated budget set except A_i. Since submodularity is the discrete analog of concavity, there is $F_i(U \cup \{a_{ij}\}) - F_i(u) \geqslant F_i(T \cup \{a_{ij}\}) - F_i(T)$ if $U \subseteq T$. Meanwhile, the term $1 - q(A_{-i}, i)$ is irrelevant with a_{ij}. Therefore, we can conclude that $w_i(\cdot)$ is submodular.

Similarly, for any other user k ($k \neq i$), the increased probability can be formulated as:

$$\begin{aligned}\delta_k(a_{ij}|A) &= (q(A_{-k} \cup \{a_{ij}\}, k) - \\ &\qquad q(A_{-k}, k))(1 - F_k(A_k)). \qquad (2)\end{aligned}$$

As shown in Claim 1, there is $\frac{\partial^2 q}{\partial b_i \partial b_j} \leqslant 0$. Accordingly, the marginal gain will be non-increasing if we add an element to a super set T, indicating $q(U \cup \{a_{ij}\}, k) - q(U, k) \geqslant q(T \cup \{a_{ij}\}, k) - q(T, k)$ if $U \subseteq T$. Therefore, the expectation $w_k(\cdot)$ is also submodular for any other user k.

Summing up the increased probabilities of all users, we have $f(A \cup \{a_{ij}\}) - f(A) \geqslant f(T \cup \{a_{ij}\}) - f(T)$ if $A \subseteq T$, showing that $f(\cdot)$ is submodular. □

For the general problem of maximizing a monotone and submodular function subject to a matroid constraint, Nemhauser *et al.*[31] showed that Algorithm 1 achieves 1/2 approximation ratio of the optimal result. Note that the approximation will still hold even when the budget is divided into unequal pieces, since the objective function still has the properties of monotonicity and submodularity. In this paper, as the budget is divided into uniform pieces, we can further improve the approximation ratio to a factor of $1 - 1/e$. Let the optimal result in the discrete setting be OPT_d. We have the following theorem.

Theorem 2. *The DiscreteGreedy budget allocation in Algorithm 1 has $1 - 1/e$ approximation ratio of the optimal result in the discrete setting, i.e., $f(A) \geqslant (1 - 1/e)OPT_d$.*

①A matroid is a pair $(X, \mathcal{I})$, where X is a finite set and $\mathcal{I}$ is a family of subsets of X with the property that if A and B are two independent sets of $\mathcal{I}$ and $B \subseteq A$, then there is an element a in A and adding a to B will give a larger independent set than B.

Algorithm 1. *DiscreteGreedy*

```
input: G, B, m
A = N × [m];
A ← ∅;
for j ← 1 to m do
    i = arg max_{s∈N}(f(A ∪ {a_sj}) − f(A));
    A ← A ∪ {a_ij};
end
Return A
```

Proof. Let $A^* = \{a_1^*, a_2^*, ..., a_m^*\}$ be the optimal solution in the discrete setting, i.e., $f(A^*) = OPT_d$. In Algorithm 1, we select A^j after j steps, meaning that $A = A^m$. Obviously,

$$\begin{aligned} f(A^*) &\leqslant f(A^* \cup A^j) \\ &= f(A^j) + \sum_{k=1}^{m}(f(A^j \cup \{a_1^*, ..., a_k^*\}) - \\ &\quad f(A^j \cup \{a_1^*, ..., a_{k-1}^*\})). \end{aligned}$$

According to the submodularity of f, when adding an element a_k^*, there is $f(A^j \cup \{a_1^*, ..., a_k^*\}) - f(A^j \cup \{a_1^*, ..., a_{k-1}^*\}) \leqslant f(A^j \cup \{a_k^*\}) - f(A^j)$. In the $(j+1)$-th step of the greedy algorithm, we choose the element from $a_{1,j+1}, a_{2,j+1}, ..., a_{n,j+1}$ which maximizes $f(A^{j+1}) - f(A^j)$. Note that in the budget allocation, the elements are uniform. Therefore, the marginal gains of adding a_{ix} and a_{iy} to A^j are the same for any $x \leqslant y \leqslant j+1$. Thus, there is $f(A^j \cup \{a_k^*\}) - f(A^j) \leqslant f(A^{j+1}) - f(A^j)$. Combining the above inequality, we have

$$\begin{aligned} f(A^*) &\leqslant f(A^j) + \sum_{a \in A^*}(f(A^{j+1}) - f(A^j)) \\ &= f(A^j) + m(f(A^{j+1}) - f(A^j)). \end{aligned}$$

The above inequality implies that $f(A^*) - f(A^{j+1}) \leqslant (1 - \frac{1}{m})(f(A^*) - f(A^j))$. By induction,

$$\begin{aligned} f(A^*) - f(A) &\leqslant (1 - \frac{1}{m})^m (f(A^*) - f(\emptyset)) \\ &= (1 - \frac{1}{m})^m f(A^*) \\ &\leqslant e^{-1} f(A^*). \end{aligned}$$

Rearranging this inequality, we yield the claim that $f(A) \geqslant (1 - 1/e)OPT_d$. □

Time Complexity. Suppose computing the marginal gain $\delta(a_{ij}|A)$ takes T_d time. The time complexity of Algorithm 1 is $O(mnT_d)$. The algorithm can run in polynomial time as long as T_d is polynomial.

However, computing the marginal gain can be intractable since computing $f(\boldsymbol{b})$ is #P-hard: given an arbitrary set $S \subset \mathcal{N}$, suppose that the utility function is $F_i(b_i) = b_i$ for each user and the budget is set as $B = |S|$. Then, for any S, we can allocate each user in S with a budget $b_i = 1$, which will cause a set of initial adopters S. In this case, computing $f(\boldsymbol{b})$ is equivalent to computing the influence of S under the IC model, which was proved to be #P-hard in [3]. Previous studies usually adopt Monte Carlo method to simulate the graph massive times (for example, 10 000) and use BFS traversals to get the average value, which takes $O(10\,000(n + |\mathcal{E}|))$ time and can be intolerable when the networks have millions of nodes. We will discuss how to scale up the algorithm in Section 4.

3.3 How to Choose m

Despite the near optimal results from Theorem 2, for more general situations, we would like to know how it approximates the optimal value in the continuous setting. In addition, the time complexity of the DiscreteGreedy algorithm may vary with the value m. In this subsection, we will discuss how to choose the discrete granularity m so that the DiscreteGreedy algorithm can reach near optimal result even in the continuous setting.

Apparently, if m decreases to 1, the algorithm is simply choosing one user to allocate. To the other extreme, when m approaches ∞, we are allocating an infinitesimal continuously in each step. In this case, the DiscreteGreedy algorithm can reach an upper bound of $1-1/e$ when $m \to \infty$. The result originates from maximizing a nondecreasing smooth submodular function.

Definition 1. *A function $g : [0,1]^X \to \mathbb{R}$ is smooth submodular*[32] *if*

• *$g \in C_2([0,1]^X)$, i.e., it has second-order partial derivatives everywhere*;

• *for any $i, j \in X$ (possibly equal), $\frac{\partial^2 g}{\partial y_i \partial y_j} \leqslant 0$ everywhere.*

Lemma 2. *The DiscreteGreedy reaches $1-1/e$ approximation ratio in the continuous setting when $m \to \infty$.*

Proof. According to Vondrák[32], the performance guarantee can be proved if the objective function f is monotone and smooth submodular. For all $i \in \mathcal{N}$, f satisfies

$$\begin{aligned} \frac{\partial f}{\partial b_i} &= \frac{\partial F_i(b_i)}{\partial b_i}(1 - q(\boldsymbol{b}_{-i}, i)) + \\ &\quad \sum_{j \neq i} \frac{\partial q(\boldsymbol{b}_{-j}, j)}{\partial b_i}(1 - F_j(b_j)) \geqslant 0, \end{aligned}$$

by the monotonicity of $F(\cdot)$ and $q(\cdot, i)$. Similarly, for

$i \neq j$, there is

$$\begin{aligned}&\frac{\partial^2 f}{\partial b_i \partial b_j}\\&= -\frac{\partial F_i(b_i)}{\partial b_i}\frac{\partial q(\boldsymbol{b}_{-i}, i)}{\partial b_j} - \frac{\partial F_j(b_j)}{\partial b_j}\frac{\partial q(\boldsymbol{b}_{-j}, j)}{\partial b_i} +\\&\quad \sum_{k\neq i, k\neq j} \frac{\partial^2 q(\boldsymbol{b}_{-k}, k)}{\partial b_i, \partial b_j}(1 - F_k(b_k))\\&\leqslant 0.\end{aligned}$$

When $i = j$, we have

$$\begin{aligned}\frac{\partial^2 f}{\partial b_i^2} &= \frac{\partial^2 F_i(b_i)}{\partial b_i^2}(1 - q(\boldsymbol{b}_{-i}, i)) +\\&\quad \sum_{j\neq i} \frac{\partial^2 q(\boldsymbol{b}_{-j}, j)}{\partial b_i^2}(1 - F_j(b_j))\\&\leqslant 0.\end{aligned}$$

Finally, the domain of f can be mapped to the range of 0 to b_i/B, which is within $[0, 1]$. Consequently, f is monotone smooth submodular and there is $f(A) \geqslant (1 - 1/e)OPT$ when $m \to \infty$. □

Even when the algorithm can reach an upper bound of $1 - 1/e$ when m approaches ∞, it requires infinite steps to get the result. To trade off between the performance and the scalability, we need to choose a proper value of m so that the DiscreteGreedy algorithm can reach constant performance guarantee and high efficiency at the same time. We show that the algorithm can reach $1 - 1/e - o(1)$ approximation ratio when $m = O(n)$, where $o(1)$ is a relative small error.

Theorem 3. *$f(A) \geqslant (1 - 1/e - o(1))OPT$ when $m = O(n)$.*

Proof. Suppose the optimal solution is a fractional vector $\boldsymbol{b}^* = (b_1^*, b_2^*, ..., b_n^*)$, $f(\boldsymbol{b}^*) = OPT$. Let $\boldsymbol{v}^*$ be a direction $\boldsymbol{v}^* = (v_1^*, v_2^*, ..., v_n^*)$, where $v_i^* = \max\{b_i^* - A_i^j, 0\}$. According to the monotonicity of f, $OPT = f(A^j + \boldsymbol{v}^*) \geqslant f(\boldsymbol{v}^*)$. Consider the ray of direction $\boldsymbol{v}^*$ starting at A, $f(A^j + \xi\boldsymbol{v}^*)$, $\xi \geqslant 0$. Since f is smooth submodular, by the mean value theorem, there exists some $c \in [0, 1]$, such that $f(A^j + \boldsymbol{v}^*) - f(A) = \frac{\mathrm{d}f}{\mathrm{d}\xi}|_{\xi=c} \leqslant \frac{\mathrm{d}f}{\mathrm{d}\xi}|_{\xi=0} = \boldsymbol{v}^* \cdot \nabla f(A^j)$②. In Algorithm 1, we choose $a_{i,j+1}$ to maximize $f(A^{j+1}) - f(A^j)$. By the smooth submodularity of f, there is

$$\frac{f(A^{j+1}) - f(A^j)}{|a_{i,j+1}|} \geqslant \nabla f(A^j \cup \{a_{i,j+1}\}).$$

By the definition of $\boldsymbol{v}^*$, there is $|v_k^*| \leqslant B = m|a_{i,j+1}|$ for any $k \in \mathcal{N}$. Thus,

$$\begin{aligned}&f(A^{j+1}) - f(A^j)\\&\geqslant \frac{1}{m}\boldsymbol{v}^*\nabla f(A^j \cup \{a_{i,j+1}\})\\&\geqslant \frac{1}{m}(f(A^j \cup \{a_{i,j+1}\} + \boldsymbol{v}^*) - f(A^j \cup \{a_{i,j+1}\}))\\&\geqslant \frac{1}{m}(OPT - f(A^j \cup \{a_{i,j+1}\})).\end{aligned}$$

Therefore, we have $OPT - f(A^{j+1}) \leqslant \frac{1}{1+\frac{1}{m}}(OPT - f(A^j))$. By induction, there is $OPT - f(A^m) \leqslant (1 - \frac{1}{1+m})^m(OPT - f(\emptyset)) \leqslant (1 - o(\frac{1}{n}))e^{-1}OPT$ when $m = O(n)$. Moving OPT to the right side, we can yield the result that $f(A) \geqslant (1 - 1/e - o(1))OPT$. □

3.4 Extensions

According to the analysis, most of the results can be extended to other diffusion models, as long as they have the same properties as stated in Claim 1. Here, we introduce another two widely used information diffusion models and show that these models also have the same properties as IC model.

- *Coverage Model.* In the Coverage model[22], a node will be active if it is an initial adopter or a neighbor of an initial adopter. Otherwise, it will be inactive. The diffusion process under Coverage model can be regarded as information exposures to users.
- *Linear Threshold (LT) Model.* In LT model[1], for each neighbor w, a node v associates a weight $\omega_{v,w} \geqslant 0$ where $\sum_{w\in N(v)} \omega_{v,w} \leqslant 1$, and chooses some threshold $\theta_v \in [0, 1]$ uniformly at random. The node v is activated at time step t if $\sum_{w\in N_t(v)} \omega_{v,w} \geqslant \theta_v$, where $N_t(v)$ denotes the neighbors of v that are active at time step t.

For both of the above diffusion models, it can be easily shown that they have the same properties as Claim 1. In the Coverage model, the diffusion function can be formulated as $q(\boldsymbol{b}_{-i}, i) = 1 - \prod_{j\in N(i)}(1 - F_i(b_i))$. As for the LT model, Kempe *et al.*[1] showed that the process of the Linear Threshold model is equivalent to the process of live-edge paths under the random selection of live edges defined as in the IC model. Thus, the diffusion function can be formulated as $q(\boldsymbol{b}_{-i}, i) = \sum_X P[X]q_X(\boldsymbol{b}_{-i}, i)$. It can be easily proved that both of the above models satisfy the properties in Claim 1.

Corollary 1. *$f(A) \geqslant (1 - 1/e - o(1))OPT$ for both the Coverage model and the LT model.*

② ∇ denotes a gradient and $\cdot$ is a dot product.

4 Scalable Algorithms

In Section 3, we have shown the approximation ratio of the DiscreteGreedy algorithm. However, the algorithm can still be computationally intensive for large networks. When allocating the budget, in each iteration, we need to compute and choose the user with the largest "marginal gain", which requires massive BFS traversals in the networks and is often time consuming. In this section, we try to scale up the DiscreteGreedy algorithm by reducing the number of BFS traversals when computing the marginal gain.

In the DiscreteGreedy algorithm, the marginal gain is the sum of increased probabilities from all related nodes. We need to compute the diffusion probability q by running massive BFS from the given node ((2)), which is time consuming. Note that the budget elements added in each iteration are uniform in our algorithm. Accordingly, to avoid many traversals, we propose to "update" the marginal gain of each node instead of "computing" the marginal gain. Given the diffusion probability $q(A, k)$, we try to update it to get $q(A \cup \{a_{ij}\}, k)$. When adding element a_{ij}, the difference between them will be

$$\begin{aligned}
& q(A \cup \{a_{ij}\}, k) - q(A, k) \\
= & \sum_{X, i \in X_k} P[X](q_X(A \cup \{a_{ij}\}, k) - q_X(A, k)) \\
= & (1 - \frac{1 - F_i(A \cup \{a_{ij}\}, k)}{1 - F_i(A_i, k)}) \\
& \sum_{X, i \in X_k} P[X](1 - q_X(A, k)).
\end{aligned}$$

Suppose $r(i, k)$ is the pairwise diffusion probability between i and k. It can also represent the probability of $i \in X_k$. Then we can estimate that $\sum_{X, i \in X} P[X](1 - q_X(A, k)) \approx r(i, k)(1 - q(A, k))$. Accordingly, the diffusion probability can be updated as:

$$\begin{aligned}
& q(A \cup \{a_{ij}\}, k) \\
= & q(A, k) + r(i, k)(1 - \frac{1 - F_i(A \cup \{a_{ij}\}, k)}{1 - F_i(A_i, k)}) \times \\
& (1 - q(A, k)).
\end{aligned}$$

Now we can update $q(A \cup \{a_{ij}\}, k)$ from $q(A, k)$ efficiently in IC model if we know the pairwise diffusion probability $r(i, k)$. In addition, we improve the algorithm using lazy-forward method[2]: as the marginal gain of adding any element is non-increasing, the element that is selected in the last iteration is still likely to get the maximal marginal gain in the current iteration. The details are presented in Algorithm 2. We use cur_i to indicate whether the marginal gain of i is updated. Since the marginal gain of each node is non-increasing, we can add a_{ij} only if $\delta(a_{ij})$ is updated and it is the maximal among all nodes; if not, we would update the marginal gain $\delta(a_{ij})$ with $q(A \cup \{a_{ij}\}, k)$. Using the "update" technique, the increased probability can be computed directly in $O(1)$ time, compared with T_d in the original DiscreteGreedy algorithm.

Algorithm 2. *DiscreteGreedy++*

```
input: G, B, m
A ← ∅;
foreach i ∈ N do δ(a_i) ← ∞;
for j ← 1 to m do
    foreach i ∈ N do cur_i ← false;
    while true do
        i = arg max_{s∈N} δ(a_s|A);
        /* Allocate a budget to i if δ(a_i|A) is updated.*/
        if cur_i then
            A ← A ∪ {a_ij};
            Break;
        end
        /* Otherwise, update the marginal gain
        δ(a_i|A).*/
        else
            foreach k ∈ N do
                Update q(A ∪ {a_ij}, k) from q(A, k);
                δ(a_i|A) ← Σ_k δ_k(a_ij|A);
                cur_i ← false;
            end
        end
    end
end
Return A
```

Next, we show how to estimate the pairwise diffusion probability $r(i, k)$ for any pair of the users in the social network. Theoretically, it is #P-hard to compute $r(i, k)$[3]. One way to get the pairwise probability is to simulate the graph massive times to get the average, which can be computationally intensive. Another possible solution is to get the probabilities offline and save the probabilities in memory. However, the space complexity is $O(n^2)$ and may not be acceptable for large networks. Note that a key step in the DiscreteGreedy++ algorithm is to update $q(A \cup \{a_{ij}\}, k)$ for every node k. It requires to get the reachability probabilities from i to every other node. Thus, we propose to use a modified BFS starting from the selected node i to get the diffusion probabilities to all reachable nodes: $r(i, 1), r(i, 2), ..., r(i, n)$. For each node, we maintain two probabilities: the lower bound diffusion probability r_l and the upper bound diffusion probability r_u. Initially, we set $r_l(i, k) = 0$ and $r_u(i, k) = 1$. The BFS starts from i. When visiting the neighbor k

of a traversed node c, we examine if the directed edge (c, k) is visited or not. If not, we can update the lower bound diffusion probabilities as

$$r_l(i,k) \leftarrow r_l(i,k)(1-\mu_{ck}(1-r_l(i,c))).$$

The upper bound diffusion probabilities can be updated as:

$$r_u(i,k) \leftarrow r_u(i,k) + \mu_{uc} r_u(i,k).$$

For the lower bound probability, the assumption is that the diffusion probability on each path is independent. And thus $r_l(i, k)$ is the negative of the probability that k is not influenced by all of the incoming edges. For the upper bound probability $r_u(i, k)$, we get the sum of the probabilities from all incoming edges iteratively. The details are presented in Algorithm 3. We compute the reachability function of i from $r(i, 1)$ to $r(i, n)$. For each neighbor k of the traversing node c in *current*, we check if the directed edge (c, k) is visited. The probabilities are updated if the edge is not visited. The algorithm returns the average of the upper and the lower probabilities.

Algorithm 3. $BFSEstimate(i)$

```
input: 𝒢, μ, i
foreach k ∈ 𝒩 do visited[k] ← false, r_u[k] ← 0,
r_l(i,k) ← 1.0;
foreach e ∈ ℰ do e_visited[e] ← false;
current ← [];
while current.size() > 0 do
    last ← [];
    for c ∈ current do
        for k ∈ N(c) do
            /*Update r(i,k).*/
            if k ≠ i and !e_visited[(c,k)] then
                r_l(i,k) ← r_l(i,k)(1 − μ_ck * (1 − r_l(i,k)));
                r_u(i,k) ← r_u(i,k) + μ_ck r_u(i,k);
                e_visited[(c,k)] ← true;
            end
            if !visited[k] then
                last.add(k);
                visited[k] ← true;
            end
        end
    end
    current ← last;
end
foreach k ∈ 𝒩 do r_l(i,k) ← 1 − r_l(i,k),
r(i,k) ← (r_l(i,k)+r_u(i,k))/2;
Return r(i,1), r(i,2), ..., r(i,n);
```

5 Evaluations

In addition to the provable performance guarantee, in this section, we conduct extensive experiments to evaluate the proposed algorithm. We first compare our algorithm with several intuitive heuristics. Then we will observe how the discrete granularity of the budget affects the results of the DiscreteGreedy++ strategy. The time efficiency of different algorithms will be discussed in the last part.

5.1 Experimental Setup

Datasets. The experiments are conducted on four real social network datasets. The first is CondMat dataset which is a collaboration network from e-print arXiv and covers scientific collaborations between papers submitted to Condense Matter category[33]. The second is a Twitter dataset collected from Twitter.com[34]. The nodes represent user accounts and the directed links represent following relationships between them. The weight of each edge indicates that a user is in multiple circles of his/her followers. The third is Youtube dataset in which the nodes are users and the edges indicate friendship relationships between them[35]. The last dataset is Weibo dataset crawled from Sina Weibo. Weibo is a Twitter-like mirco-blog in China[36]. The directed edge (i, j) means that user i is following j. All four datasets exhibit small world and high clustering complex network structural features. The statistics of the datasets are listed in Table 2.

Utility Functions. Since our algorithm is applicable for any form of concave functions of user utilities, we manually set the distribution functions in the experiments. We first consider a uniform symmetric setting in which $F_i(b_i) = \frac{b_i}{\tau}$ with $\tau = 5$. The user utility is distributed uniformly in the range $[0, \tau]$. The second is a differential utility function: $F_i(b_i) = \frac{r+d_i+1}{r+b_i} \frac{b_i}{d_i+1}$ with $r = 10$. The threshold here is $d_i + 1$. This function has a very deep increase in the shape at the beginning. The last one is $F_i(b_i) = \sqrt{\frac{b_i}{\tau}}$ with $\tau = 5$. The function also has a threshold τ and the utility is proportional to the square root of the allocated budget.

Diffusion Models. For the information diffusion models, since all of them exhibit monotone submodular properties, we mainly observe the results in IC and Coverage model. In the IC model, as the information diffusion process under this model is a stochastic process, we need to take Monte Carlo methods to simulate the graph massive times (10 000) and take the average influence of different results. Without loss of generality, we first assume uniform diffusion probability μ_e on each edge and assign μ_e to be 0.01 for all $e \in \mathcal{E}$. We also consider non-uniform diffusion probabilities. In

Table 2. Datasets

Network	Number of Nodes	Number of Edges	Directed	Max. Degree	Avg. Degree	m
CondMat	23 133	93 497	Undirected	280	8.01	10 000
Twitter	81 306	1 768 135	Directed	3 383	21.75	10 000
Youtube	567 445	1 976 329	Undirected	26 632	6.97	100 000
Weibo	877 391	1 419 850	Directed	58 602	1.43	100 000

the CondMat dataset, we select the diffusion probabilities uniformly from $\{0.01, 0.05, 0.1\}$. In the Twitter dataset, since each user can list his/her friends in different ego-networks, we set the diffusion probability μ_{uv} as $0.01\#ego$ if user v appears in $\#ego$ times of v's ego-networks. In Coverage model, the result can be more straightforward since the influence of user i is the set of users reachable from the current user.

Comparison Methods. In our DiscreteGreedy++ algorithm, we discretize the budget into approximately $O(n)$ pieces for budget allocation to obtain near optimal solutions. The detail number of m is listed in Table 2. We compare the algorithm with the following intuitive heuristics.

• *Uniform.* The Uniform budget allocation adopts a simple idea that all users get the same budget $b_i = \frac{B}{n}$, regardless of users' influence and utility.

• *Proportional.* In the Proportional budget allocation, the budget allocated to user i is proportional to d_i, i.e., $b_i = \frac{d_i}{\sum_j d_j} B$.

• *PageRank.* The PageRank first assigns uniform budget to each user and then uses random walk with restart to get the rank value for each user. The maximum iteration number is set as 100 000 and the minimum delta is 0.000 01. The restart probability is 0.2.

• *FullGreedy.* We use the greedy algorithm in [1] to select the initial adopters and offer each of them the threshold utility. The threshold is τ for the first two utility functions and $d_i + 1$ in the third function. Note that the threshold may not exist for some other utility functions.

• *DiscreteGreedy.* We also implement the origin DiscreteGreedy algorithm before using the scaling-up techniques. In particular, we also adopt the existing CELF idea in the DiscreteGreedy algorithm.

5.2 Results

Spread of Viral Advertising. In Fig.5∼Fig.7, we separately present the results w.r.t. different utility functions under the IC model and the Coverage model. The x-axis represents the total budget that is allocated. And the y-axis is the active set size of the users after cascade, or the spread of the viral advertising.

In Fig.5, we fix the utility function as $F_i(b_i) = \frac{b_i}{\tau}$ and conduct the experiments under the IC model in CondMat and Twitter datasets. In this case, since the utility function grows linearly to the threshold τ, the marginal gain of a user will remain the same in the range $b_i \in (0, \tau]$ ($\frac{\partial^2 f}{\partial b_i^2} = 0$). Therefore, the FullGreedy algorithm has the same result as the DiscreteGreedy algorithm, and will not be presented here. The diffusion probability on each edge is set uniformly as 0.01 in Fig.5(a) and Fig.5(b). As presented, the results of DiscreteGreedy and DiscreteGreedy++ are almost the same, showing that the BFSEstimate algorithm (Algorithm 3) can achieve high accuracy when computing the pairwise diffusion probabilities. And both of them perform significantly better than other algorithms. The Proportional and the PageRank algorithms have relatively better performance in the Twitter dataset than CondMat, since the network is denser and nodes degree plays a more important role in the diffusion. We then select μ_e from $\{0.01, 0.05, 0.1\}$ or proportional to $\#ego$ in Fig.5(c) and Fig.5(d). In this case, the random graph is likely to have a giant component. And the algorithms can reach high performance even with small budget.

Fig.6 presents the results with the utility function as $F_i(b_i) = \frac{r+d_i+1}{r+b_i} \frac{b_i}{d_i+1}$ and $r = 10$ in IC model. In this function, different users have different utility distributions, where the utility is a function of users' degrees. For a user with degree d_i, the threshold will be $d_i + 1$. Moreover, this utility function has a steep increase at the beginning. As before, the DiscreteGreedy++ has almost the same performance with the DiscreteGreedy algorithm. The FullGreedy algorithm could have quite poor performance since the marginal gain of each user decreases very quickly as the allocated budget grows. Note that the threshold is actually not necessary in practice. The FullGreedy algorithm may be infeasible if users do not have a threshold. The Uniform algorithm can be more ineffective in larger Twitter dataset due to the low budget offered to each user.

Finally, in Fig.7, we set the utility function as $F_i(b_i) = \sqrt{\frac{b_i}{\tau}}$ and conduct the experiments under the

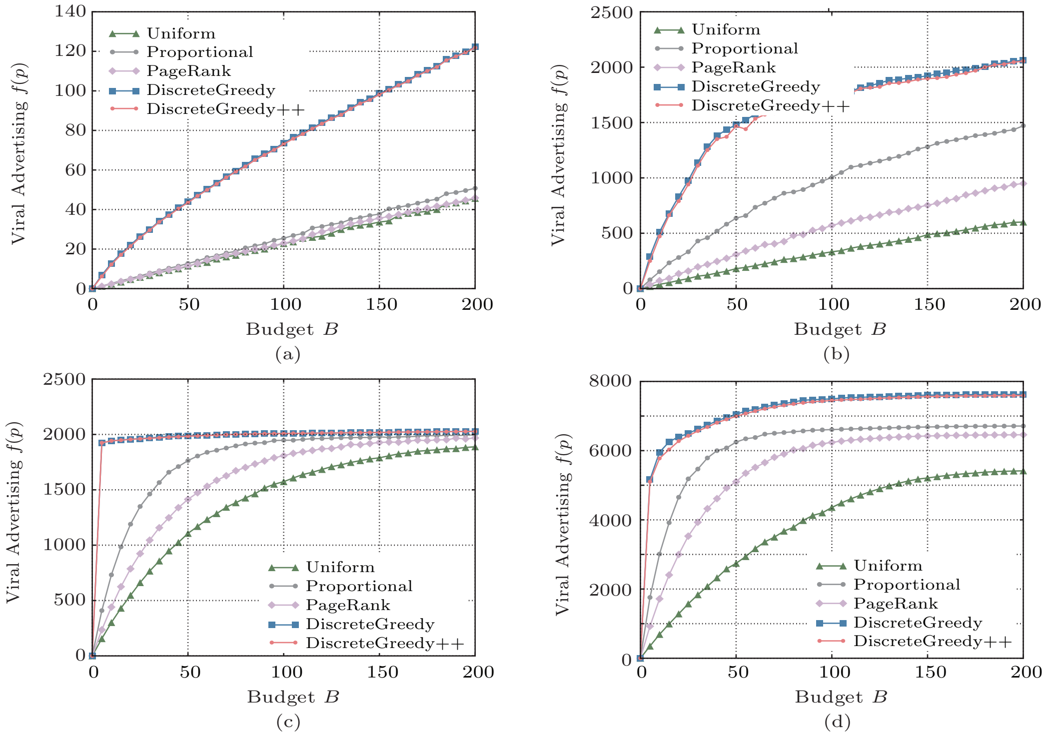


Fig.5. Spread of advertising in the IC model with utility function $F_i(b_i) = \frac{b_i}{\tau}$. (a) CondMat with $\mu_e = 0.01$. (b) Twitter with $\mu_e = 0.01$. (c) CondMat with μ_e from $\{0.01, 0.05, 0.1\}$. (d) Twitter with $\mu_e = 0.01\#ego$.

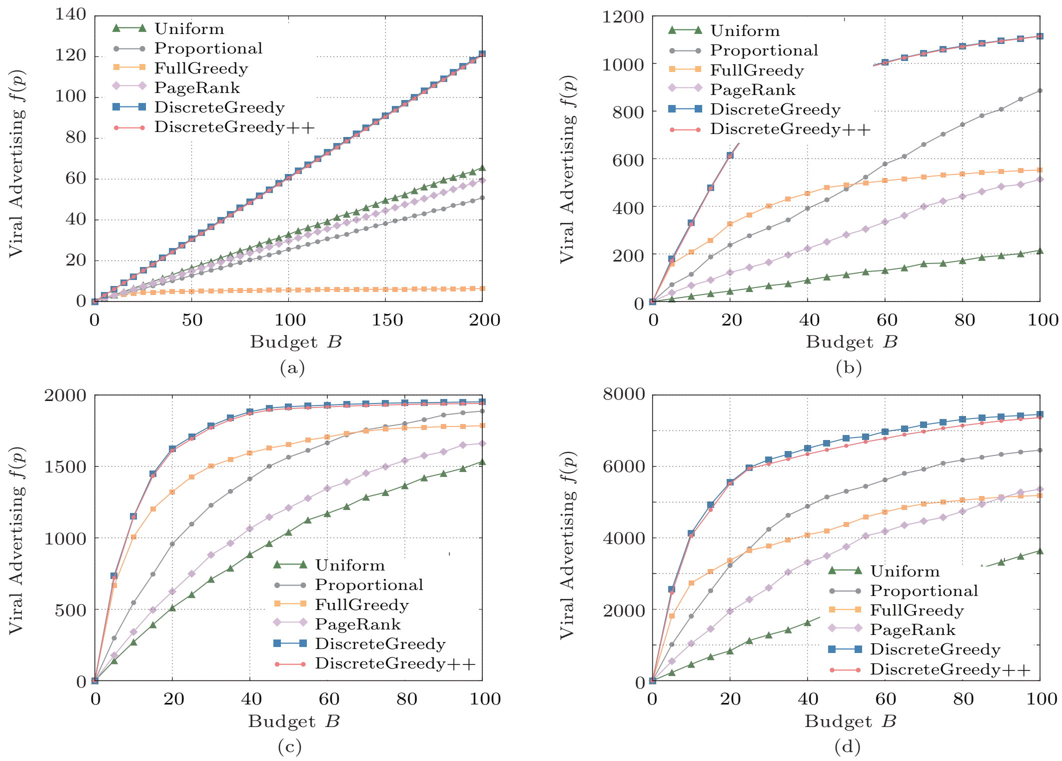


Fig.6. Spread of viral advertising in the IC model with utility function $F_i(b_i) = \frac{r+d_i+1}{r+b_i}\frac{b_i}{d_i+1}$. (a) CondMat with $\mu_e = 0.01$. (b) Twitter with $\mu_e = 0.01$. (c) CondMat with μ_e from $\{0.01, 0.05, 0.1\}$. (d) Twitter with $\mu_e = 0.01\#ego$.

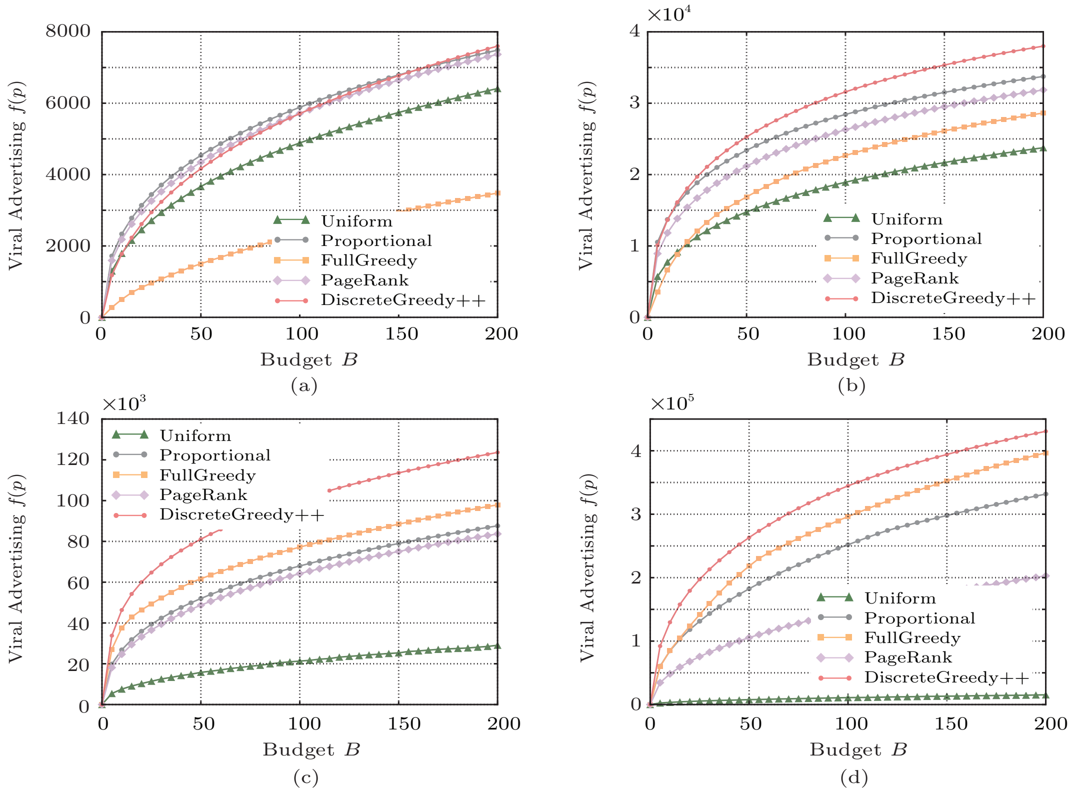


Fig.7. Spread of viral advertising in the Coverage model with utility function $F_i(b_i) = \sqrt{\frac{b_i}{\tau}}$. (a) CondMat. (b) Twitter. (c) Youtube. (d) Weibo.

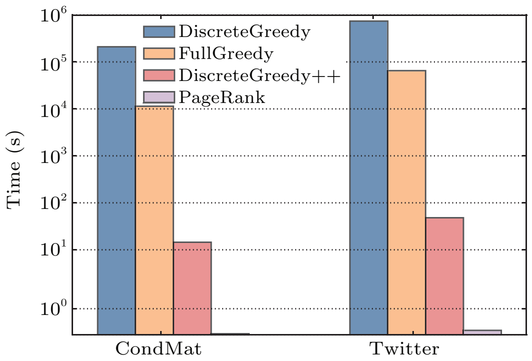
Coverage model. Obviously in all cases, our DiscreteGreedy++ algorithm outperforms other heuristics significantly, showing that the algorithm can also be extended to the Coverage model. Since the marginal gain of each user decays with the allocated budget, the FullGreedy performs worse than the DiscreteGreedy++. The Proportional and the PageRank algorithm have similar performance since both of them allocate more budget to the users with higher centrality. The Uniform algorithm has relatively better results in the small datasets like CondMat, since the users can be allocated more budget for being initial adopters. However, when the size of the networks grows, the performance of Uniform algorithm can be quite poor especially in the Weibo dataset.

Accuracy. In this experiment, we run the DiscreteGreedy++ algorithm in CondMat dataset w.r.t. different granularities in the Coverage model. The budget is discretized into m pieces where m ranges from 1 to $10O(n) = 100\,000$. The results are presented in Table 3. As presented, the spread of viral advertising converges when m approaches $O(n) = 10\,000$, i.e., $n/m \approx 1$. When user utility functions are linear, i.e., when $F_i(b_i) = \frac{b_i}{\tau}$, the DiscreteGreedy++ algorithm would allocate the selected users threshold utility. Therefore, the results can converge when the granularity is the threshold τ. In other cases, though the active set size converges slower, the results can stabilize when $m \to O(n)$. Even when m grows higher to $10O(n) = 100\,000$, the results are not likely to grow much.

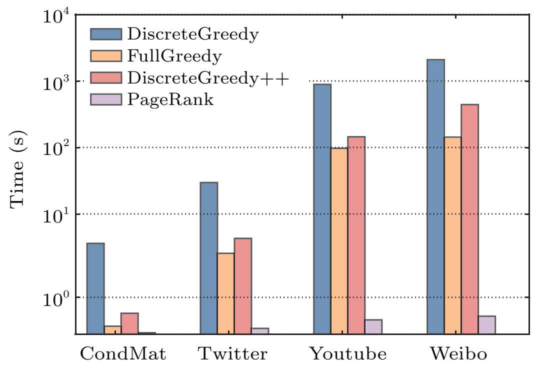
Table 3. Viral Advertising in CondMat Dataset w.r.t. Different Granularities

m	F_i		
	$\frac{b_i}{\tau}$	$\frac{\tau+d_i+1}{\tau+b_i}\frac{b_i}{d_i+1}$	$\sqrt{\frac{b_i}{\tau}}$
1	281.00	277.14	281.00
10	3 309.24	1 541.26	3 620.32
100	3 452.58	1 605.29	5 296.93
1 000	3 483.00	1 608.26	6 939.77
10 000	**3 483.00**	**1 608.25**	**7 599.19**
100 000	3 483.00	1 608.27	7 600.15

Scalability. Finally we analyze the scalability of the algorithms in different datasets. Specifically, we compare the efficiency of the DiscreteGreedy and the DiscreteGreedy++ algorithm before and after using the scaling-up techniques introduced in Section 4. The algorithms are running on a Linux server with a 2.66 GHz i7 CPU and 6 Gb RAM in C++ code. We also adopt lazy forward methods[2] for the DiscreteGreedy and the FullGreedy algorithms. The results of Uniform and Proportional are not presented since both of them can be finished in $O(1)$ time. Fig.8 shows the running time of different algorithms in the IC and the Coverage model. As expected, DiscreteGreedy++ takes much less time than DiscreteGreedy, since we adopted scaling-up techniques in the DiscreteGreedy++ algorithm. DiscreteGreedy++ can be very efficient under the Coverage model even with millions of nodes. For the IC model, the consumed time is still within an acceptable range. Even when there is a large gap between the granularity of the DiscreteGreedy++ and the FullGreedy algorithms, the running time is almost the same. Among the algorithms, PageRank achieves the highest efficiency. However, the performance may be too poor for usage.



(a)



(b)

Fig.8. Running time of different algorithms in the IC and the Coverage model. (a) IC model. (b) Coverage model.

6 Conclusions

In this work, we studied budget allocation problem in a social network to maximize viral advertising with budget constraint. Compared with the classical influence maximization problem which aims to select the most influential set of users, we proposed that user utilities for being initial adopters are often probabilistic rather than deterministic. In particular, the user utilities can be modeled as non-decreasing concave functions. Suppose the viral advertising can spread in the social network according to the IC model. We formalized the budget allocation as an optimization problem and established its hardness. A novel DiscreteGreedy++ algorithm with near optimal results was proposed with performance guarantee, and scaling-up methods were introduced for higher efficiency. Extensive evaluations showed that our DiscreteGreedy++ algorithm outperforms other intuitive heuristics significantly in almost all cases.

For possible future work, we are interested in the following aspects. First, we would like to learn the user utility functions for being initial adopters empirically. The online social networks have provided rich historical datasets of user behaviors and social interactions. Thus, it is possible for us to infer user utilities accurately. Second, even when the advertising can spread in the social network according to the diffusion models, the diffusion probabilities often depend on the content of the advertising. We would try to encourage users to composite advanced contents. For example, the users can get incentives proportional to their contribution to the viral advertising.

References

[1] Kempe D, Kleinberg J, Tardos É. Maximizing the spread of influence through a social network. In *Proc. the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug. 2003, pp.137-146.

[2] Leskovec J, Krause A, Guestrin C, Faloutsos C, VanBriesen J, Glance N. Cost-effective outbreak detection in networks. In *Proc. the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug. 2007, pp.420-429.

[3] Chen W, Wang Y, Yang S. Efficient influence maximization in social networks. In *Proc. the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Jun. 2009, pp.199-208.

[4] Demaine E D, Hajiaghayi M, Mahini H, Malec D L, Raghavan S, Sawant A, Zadimoghadam M. How to influence people with partial incentives. In *Proc. the 23rd International Conference on World Wide Web*, April 2014, pp.937-948.

[5] Tang Y, Xiao X, Shi Y. Influence maximization: Near-optimal time complexity meets practical efficiency. In *Proc. the 2014 ACM SIGMOD International Conference on Management of Data*, Jun. 2014, pp.75-86.

[6] Marshall A. Principles of Economics: Unabridged Eighth Edition (8th edition). Cosimo, Inc., 2009.

[7] Ioannidis S, Chaintreau A, Massoulié L. Optimal and scalable distribution of content updates over a mobile social network. In *Proc. IEEE INFOCOM*, Apr. 2009, pp.1422-1430.

[8] Richardson M, Domingos P. Mining knowledge-sharing sites for viral marketing. In *Proc. the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Jul. 2002, pp.61-70.

[9] Domingos P, Richardson M. Mining the network value of customers. In *Proc. the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug. 2001, pp.57-66.

[10] Lou T, Tang J. Mining structural hole spanners through information diffusion in social networks. In *Proc. the 22nd International Conference on World Wide Web*, May 2013, pp.825-836.

[11] Chen W, Wang C, Wang Y. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *Proc. the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Jul. 2010, pp.1029-1038.

[12] Chen W, Yuan Y, Zhang L. Scalable influence maximization in social networks under the linear threshold model. In *Proc. the 10th IEEE International Conference on Data Mining* (*ICDM*), Dec. 2010, pp.88-97.

[13] Hartline J, Mirrokni V, Sundararajan M. Optimal marketing strategies over social networks. In *Proc. the 17th International Conference on World Wide Web*, Apr. 2008, pp.189-198.

[14] Chen W, Lu P, Sun X, Tang B, Wang Y, Zhu Z A. Optimal pricing in social networks with incomplete information. In *Proc. the 7th Int. Workshop on Internet and Network Economics*, Dec. 2011, pp.49-60.

[15] Arthur D, Motwani R, Sharma A, Xu Y. Pricing strategies for viral marketing on social networks. In *Proc. the 5th Int. Workshop on Internet and Network Economics*, Dec. 2009, pp.101-112.

[16] Singer Y, Mittal M. Pricing mechanisms for crowdsourcing markets. In *Proc. the 22nd International Conference on World Wide Web*, May 2013, pp.1157-1166.

[17] Bloch F, Quérou N. Pricing in networks. Technical Report, Ecole Polytechnique, 2008. https://www.econbiz.de/Record/pricingin-networks-bloch-francis/10008794143, May 2016.

[18] Candogan O, Bimpikis K, Ozdaglar A. Optimal pricing in networks with externalities. *Operations Research*, 2012, 60(4): 883-905.

[19] Akhlaghpour H, Ghodsi M, Haghpanah N, Mirrokni V S, Mahini H, Nikzad A. Optimal iterative pricing over social networks. In *Proc. the 6th Int. Workshop on Internet and Network Economics*, Dec. 2010, pp.415-423.

[20] Lu W, Lakshmanan L V. Profit maximization over social networks. In *Proc. the 12th IEEE International Conference on Data Mining* (*ICDM*), Dec. 2012, pp.479-488.

[21] Zhu Y, Lu Z, Bi Y, Wu W, Jiang Y, Li D. Influence and profit: Two sides of the coin. In *Proc. the 13th IEEE International Conference on Data Mining* (*ICDM*), Dec. 2013, pp.1301-1306.

[22] Singer Y. How to win friends and influence people, truthfully: Influence maximization mechanisms for social networks. In *Proc. the 5th ACM International Conference on Web Search and Data Mining*, Feb. 2012, pp.733-742.

[23] Doo M, Liu L. Probabilistic diffusion of social influence with incentives. *IEEE Trans. Services Computing*, 2014, 7(3): 387-400.

[24] Yang Y, Mao X, Pei J, He X. Continuous influence maximization: What discounts should we offer to social network users? In *Proc. ACM SIGMOD*, June 26-July 1, 2016. (to be appeared)

[25] Manski C F. Maximum score estimation of the stochastic utility model of choice. *Journal of Econometrics*, 1975, 3(3): 205-228.

[26] Zadimoghaddam M, Roth A. Efficiently learning from revealed preference. In *Proc. the 8th Int. Workshop on Internet and Network Economics*, Dec. 2012, pp.114-127.

[27] Anagnostopoulos A, Kumar R, Mahdian M. Influence and correlation in social networks. In *Proc. the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug. 2008, pp.7-15.

[28] Cha M, Haddadi H, Benevenuto F, Gummadi P K. Measuring user influence in twitter: The million follower fallacy. In *Proc. the 4th ICWSM*, May 2010.

[29] Lin S, Wang F, Hu Q, Yu P S. Extracting social events for learning better information diffusion models. In *Proc. the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug. 2013, pp.365-373.

[30] Bakshy E, Karrer B, Adamic L A. Social influence and the diffusion of user-created content. In *Proc. the 10th ACM Conference on Electronic Commerce*, Jul. 2009, pp.325-334.

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

[32] Vondrak J. Optimal approximation for the submodular welfare problem in the value oracle model. In *Proc. the 40th Annual ACM Symposium on Theory of Computing*, May 2008, pp.67-74.

[33] Leskovec J, Kleinberg J, Faloutsos C. Graph evolution: Densification and shrinking diameters. *ACM Trans. Knowledge Discovery from Data* (*TKDD*), 2007, 1(1): Article No. 2.

[34] McAuley J J, Leskovec J. Learning to discover social circles in ego networks. In *Proc. Advances in Neural Information Processing Systems*, Dec. 2012, pp.548-558.

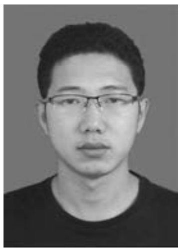
[35] Mislove A E. Online social networks: Measurement, analysis, and applications to distributed information systems [Ph.D. Thesis]. Rice University, Houston, 2009.

[36] Fu K W, Chan C H, Chau M. Assessing censorship on microblogs in China: Discriminatory keyword analysis and the real-name registration policy. *IEEE Internet Computing*, 2013, 17(3): 42-50.

Bo-Lei Zhang received his B.S. degree in computer science from Nanjing University, Nanjing, in 2006. He is a Ph.D. candidate in the Department of Computer Science & Technology and the State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing. His research interests include information diffusion and influence analysis in social networks.

Zhu-Zhong Qian received his Ph.D. degree in computer science from Nanjing University, Nanjing, in 2007. He is an associate professor in the Department of Computer Science & Technology and the State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing. His current research interests include cloud computing, distributed systems, and pervasive computing. He has published more than 30 research papers in related fields.

Wen-Zhong Li received his Ph.D. degree in computer science from Nanjing University, Nanjing, in 2007. He is an associate professor in the Department of Computer Science & Technology and the State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing. His research interests include wireless networks, pervasive computing, and social networks. He has published more than 40 research papers in international conferences and journals, including INFOCOM, ICDCS, IWQoS, ICPP, ICC, WCNC, IEEE TPDS, IEEE TWC, IEEE TVT, etc.



Bin Tang received his Ph.D. degree in computer science from Nanjing University, Nanjing, in 2014. He is currently an assistant researcher in the Department of Computer Science & Technology and the State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing. His current research interests lie in the area of communications, network coding, and distributed computing, with a focus on the application of network coding to data dissemination in various networking environments.

Sang-Lu Lu received her B.S., M.S., and Ph.D. degrees in computer science from Nanjing University, Nanjing, in 1992, 1995, and 1997 respectively. She is currently a professor in the Department of Computer Science & Technology and the State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing. Her research interests include distributed computing, wireless networks, and pervasive computing.

Xiaoming Fu received his Ph.D. degree from Tsinghua University, Beijing, in 2000. He was a research staff member with the Technical University Berlin, Berlin, Germany. Since 2002, he has been with the University of Göttingen, Göttingen, Germany, where he became a professor in 2007. He is the coordinator of the European Union Seventh Framework Programme GreenICN and MobileCloud Projects. He is the author of over 100 papers in journals and international conference proceedings. His research interests include network architectures, protocols, and applications. Dr. Fu is currently an editorial board member of the IEEE Communications Magazine, the IEEE Transactions on Network and Service Management, Elsevier Computer Networks, and Computer Communications. He received the 2005 University of Götingen Foundation Award for Exceptional Publications by Young Scholars and the Best Paper Award at the 2013 IEEE International Workshop on Local and Metropolitan Area Networks.