Neighborhood Matters: Influence Maximization in Social Networks with Limited Access

Chen Feng, Luoyi Fu, Bo Jiang, Haisong Zhang, Xinbing Wang, Feilong Tang and Guihai Chen

Abstract—Influence maximization (IM) aims at maximizing the spread of influence by offering discounts to influential users (called seeding). In many applications, due to user's privacy concern, overwhelming network scale etc., it is hard to target any user in the network as one wishes. Instead, only a small subset of users is initially accessible. Such access limitation would significantly impair the influence spread, since IM often relies on seeding high degree users, which are particularly rare in such a small subset due to the power-law structure of social networks.

In this paper, we attempt to solve the limited IM in real-world scenarios by the adaptive approach with seeding and diffusion uncertainty considered. Specifically, we consider fine-grained discounts and assume users accept the discount probabilistically. The diffusion process is depicted by the independent cascade model. To overcome the access limitation, we prove the set-wise friendship paradox (FP) phenomenon that neighbors have higher degree in expectation, and propose a two-stage seeding model with the FP embedded, where neighbors are seeded. On this basis, for comparison we formulate the non-adaptive case and adaptive case, both proven to be NP-hard. In the non-adaptive case, discounts are allocated to users all at once. We show the monotonicity of influence spread w.r.t. discount allocation and design a two-stage coordinate descent framework to decide the discount allocation. In the adaptive case, users are sequentially seeded based on observations of existing seeding and diffusion results. We prove the adaptive submodularity and submodularity of the influence spread function in two stages. Then, a series of adaptive greedy algorithms are proposed with constant approximation ratio. Extensive experiments on real-world datasets show that our adaptive algorithms achieve larger influence spread than non-adaptive and other adaptive algorithms (up to a maximum of 116%).

Index Terms—Influence maximization, access limitation, adaptive approach.

✦

1 INTRODUCTION

THE last two decades have witnessed the dramatic development of social networks (e.g., Facebook, Twitter), which have become an important platform for the promotion of ideas, behaviors and products. For example, viral marketing is a widely adopted strategy in the promotion of new products. The company selects some users and provides them with some discounts within a predefined budget, hoping that the product will be known by more users via the "word-of-mouth" effect. This demand naturally raises the influence maximization problem, which aims at triggering the largest cascade of influence by allocating discounts to users (called seeding). Since the seminal work of Kempe *et al.* [1], numerous efforts have been made to advance the research of the influence maximization problem [2][3] [4] [5] [6] [7] [8] [9].

In most cases, the problem is solved under the implicit assumption that the company can allocate discounts to any user in the network. However, in the real-world commercial campaign, the company often only has access to a small subset of users. For example, an online merchant wishes to promote a new product by providing samples to influential

users. In practice, the merchant could only mail samples to customers who have left address information before in ways such as buying products, applying for membership. Similarly, in many other applications, due to privacy concern, network scale etc., the seeding process is limited to a small sample of the network (like most work, the network structure and the diffusion probabilities are assumed pre-known and are not of concern). Due to the power-law degree distribution of social networks, high degree users are particularly rare in the small subset. Since influence maximization often relies on seeding many high degree users (not necessarily the highest ones), the access limitation would evidently impair the influence spread. Regarding this concern, initial attempts [10] [11] [12] [13] seed the subset of users to reach neighbors who are voluntary to join in the campaign automatically with the intuition of the friendship paradox (FP) phenomenon [14] which reveals that the degree of your neighbor is greater than yours in expectation. As pioneers in the access limitation problem, [10] [11] [12] [13] have largely expanded the influence spread but still fall short of dealing with the uncertainties in real promotions. Such uncertainties include two aspects. (1) *Seeding uncertainty*: a targeted user will not necessarily become the seed if the discount is not satisfactory. (2) *Diffusion uncertainty*: due to the strength of social relationships or characteristics of users, the influence propagation between two users is not assured to be successful. Existing works distribute discounts to users all at once, without considering whether the actual seeding and diffusion is successful. Such fixed strategy (referred to as "non-adaptive" method) is vulnerable to the uncertainty in the seeding and diffusion process, resulting in

unsatisfactory influence spread. Thus, we are motivated to study an adaptive [1] approach, where users are sequentially seeded based on previous seeding and diffusion results.

Specifically, the problem is investigated under the following settings. Suppose a company wants to promote a new product through a social network by providing discounts for users. Due to the difficulty in collecting user's information, only a small subset of users is initially accessible, denoted as X. We consider a fine-grained discount setting, i.e., discounts take value from $[0, 1]$ instead of only 0 or 1 in previous works on this problem. Accordingly, a user probabilistically accepts the discount and becomes a seed, from which the diffusion starts. The diffusion process is depicted by the widely acknowledged independent cascade model. Since social networks leave traces of behavioral data which allow observing and tracking, the spread of influence could be easily observed. For example, from one's social account, we can see whether the user adopts the product. On this basis, we make the first attempt to solve the limited influence maximization problem by the adaptive approach. Moreover, we investigate the non-adaptive method under this setting for comparison. When undertaking this study, we find it is challenging in the following three aspects.

Access Limitation: Under our setting, the influence spread may suffer more from the access limitation due to user's uncertain nature. Thus, there is an increasing demand on a new seeding model to address the access limitation with user's probabilistic behavior considered. We attempt to design an effective model with natural intuition and theoretical support.

Fine-grained Discount: With a larger discount space, the scale of possible discount allocations accordingly becomes greater in order sense. Thus, it is much harder to find an effective discount allocation. Meanwhile, the corresponding seeding uncertainty imposes additional difficulties on influence analysis and algorithm design.

Algorithm Design: In our seeding model, initially accessible users are seeded to reach their neighbors for further discount allocation. It is easy to see that the two seeding processes are inter-related. This interdependency requires our algorithm to collectively consider the two seeding processes. Not only seeding results of initial users should be considered but also possible seeding results of neighbors.

To overcome the access limitation, we intend to leverage the FP phenomenon that neighbors have higher degree in expectation. A new seeding model is proposed with the FP embedded. We first seed users in X (stage 1) to reach their neighbors for further discount allocation (stage 2). Whether a user u accepts the discount c_u is depicted by the seed probability function $p_u(c_u)$. Accordingly, we formulate non-adaptive and adaptive cases, both proven to be NP-hard. (1) *Non-adaptive case*: discounts are allocated to users in X all at once and then neighbors of those who accept the discount. (2) *Adaptive case*: we sequentially seed users in X by adopting actions, defined as user-discount pairs, based on previous seeding and diffusion results. Each time, if the user

accepts the discount, we further seed his/her neighbors. The main contributions of this paper are highlighted as follows.

- We first formulate the limited IM problem under fine-grained discounts and uncertain user nature. Then, we look into the FP phenomenon in the set of users X and prove it to hold set-wisely in any network. With this theoretical support, we are inspired to design a two-stage seeding model where neighbors of X are seeded, and thus we get access to more influential users and the influence spread is also expanded.

- In the non-adaptive case, we first show the monotonicity of influence spread w.r.t. the discount allocation. Then, a two-stage coordinate descent framework is designed to decide the fine-grained discount in two stages. We collectively consider the seeding process in stage 2 when designing the discount allocation of stage 1.

- In each round of the adaptive case, we calculate the benefit of each action (user-discount pair) by estimating the increase of the influence spread when neighbors of the user are seeded. Then, we adopt the action with the largest benefit-to-discount ratio. With this idea, two algorithms π^{greedy} and $\pi^{\text{greedy}}_{\text{discrete}}$ are devised with performance guarantee, where discounts of neighbors are determined by the coordinate descent and greedy algorithm respectively. Furthermore, $\pi^{\text{greedy}}_{\text{enum}}$ is proposed with the enumeration idea, achieving an approximation ratio of $(1 - e^{-\frac{B_1 - 1}{B_1}(1 - \frac{1}{e})}) \approx 0.469$ better than previous work [2] [10] [12] [13].

- Besides theoretical guarantees, our algorithms also exhibit favorable performances in experiments. Specifically, extensive experiments are conducted on real-world social networks to evaluate the proposed algorithms. The results show that our adaptive algorithms achieve much larger influence spread than non-adaptive and other adaptive algorithms (up to a maximum of **116%**), and meanwhile are scalable to large networks.

The rest of the paper is organized as follows. Important milestones are reviewed in Section 2. We describe the model in Section 3. The non-adaptive and adaptive cases are formulated in Section 4. In Section 5, we analyze the non-adaptive case. And the adaptive case is studied in Section 6. Numerous experiments are conducted in Section 7. And we conclude the paper in Section 8. Due to space limitation, some technical proofs and experimental results are deferred to the supplemental file.

2 RELATED WORK

Domingos and Richardson [4] took the head in exploring the peer influence among customers by modeling markets as social networks. They attempted to maximize the cascade of purchasing behavior by targeting a subset of users. Kempe *et al.* [1] further formulated the well-known influence maximization problem. The greedy algorithm is proposed with proven performance guarantee $(1 - e^{-1})$ [15]. No polynomial time algorithm could have a better performance, unless $P = NP$ [16]. Since then, extensive works have been done in different perspectives [17] [3] [5] [18] [19].

---

1. We mean "adaptive" in the sense that users are sequentially seeded based on previous seeding and diffusion results, while the concept in [10] [11] [12] [13] means that the allocation in neighbors is related to the set of seeds in initially accessible users.

2. Although [11] achieves an approximation ratio of $1 - \frac{1}{e}$, its diffusion model, called the voter model, is much simpler where the influence of each user is simply additive and could be anticipated in advance.

Bharathi *et al.* analyzed the game of competing information diffusions in one social network [2]. Li *et al.* considered location information in influence maximization [20]. Tang *et al.* presented the near-optimal time algorithm to solve the influence maximization problem for triggering models without hurting the approximation ratio [21]. Estimation techniques are applied in [7] to improve the empirical efficiency. Yang *et al.* [8] assumed that the discount can be fractional instead of only 0 or 1. Chen *et al.* [22] improved the influence maximization in both running time and cascade size. Budgeted influence maximization was studied in [6] where each user is associated with a cost for selection. A synthetic survey is provided in [23].

However, the access limitation is ignored by all the above works. Seeman *et al.* [13] studied the seed selection based on the intuition of FP, which is first discovered by Feld [14] and further investigated point-wisely by numerous works [24] [25] [26]. Badanidiyuru *et al.* considered monotone submodular objective functions and achieve a $(1 - e^{-1})^2$ approximation ratio [10]. In [11], by relaxing the diffusion model and discount setting, Horel *et al.* made impressive progress on algorithm design and experimental validation.

Adaptive seeding is an emerging topic in influence maximization. Users are seeded one after another based on the existing seeding and diffusion results. Golovin *et al.* [27] studied the adaptive submodularity and showed that a greedy policy obtains a $(1 - e^{-1})$ approximation ratio. Yuan and Tang proposed the adaptive algorithm based on the greedy policy [9]. In [28], only partial feedback is observed before seeding the next user.

Our work is distinct from existing works [10] [11] [12] [13] mainly in two aspects. First, the problem is comprehensively studied under practical settings, where users accept the fine-grained discounts probabilistically and the diffusion process is depicted by the well-received independent cascade model. On this basis, algorithms are proposed with theoretical guarantee and evaluated on real-world datasets. Second, all existing solutions to the access limitation can be classified as non-adaptive category. However, not only non-adaptive solution but also adaptive solution is presented in our paper to maximize the influence spread under access limitation.

3 MODEL

A social network is denoted as the graph $G(V, E)$, where V is the set of users and E records the relationships between users. Initially accessible users are denoted as $X \subseteq V$. For any user $u \in V$, let $N(u)$ denote the neighborhood of u. For any subset of users $T \subseteq V$, $N(T)$ represents the neighborhood of T, defined as $N(T) \equiv \bigcup_{u \in T} N(u) \setminus T$.

We start the influence diffusion by allocating discounts to users, which is described by the two-stage seeding model. If a user accepts the discount allocated, we say it becomes a *seed*. A set of seeds forms a *seed set*. Especially, we call the seed in stage 1 an *agent*. The diffusion process is described by the independent cascade model.

The two-stage seeding model contains *the recruitment stage* (stage 1) and *the trigger stage* (stage 2). The predefined budget is B_1 in stage 1 and B_2 in stage 2 [3], with the total budget being $B = B_1 + B_2$. The seeding process in the two-stage model is shown as follows.

- *the recruitment stage*: We seed users in X with given budget B_1, to recruit agents whose friends are possibly more influential. Some users become the agent and bring their friends (newly reachable users) into this campaign in the meantime by forwarding the promotion link, providing address information for mail, even helping hand over the sample, etc..
- *the trigger stage*: We seed newly reachable users with budget B_2 to trigger the largest influence cascade.
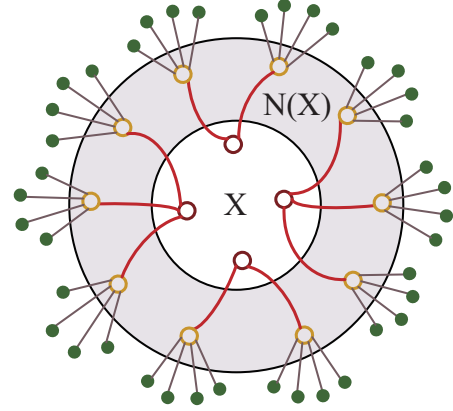

Fig. 1. An example of the two-stage seeding model. We first seed users in X and then reach their neighbors $N(X)$ for further seeding.

The discount allocated to user u is denoted as c_u. Whether user u accepts the discount is depicted by the seed probability function $p_u(c_u)$. We assume $p_u(\cdot)$ satisfies the four properties: (1) $p_u(0) = 0$; (2) $p_u(1) = 1$; (3) $p_u(\cdot)$ is monotonically nondecreasing; (4) $p_u(\cdot)$ is continuously differentiable.

In both stages, we allocate the budget to seed users, but the intention is different. In *the recruitment stage*, users are seeded to recruit agents such that we can access their influential neighbors. In *the trigger stage*, where the influence diffusion really begins, we seed the newly reached neighbors to maximize the influence diffusion. In classical influence maximization, we are restricted to seed users only in X without the privilege to reach more influential users, resulting in limited cascade size.

The diffusion model we adopt is the *independent cascade model*, which has been one of the most popular models since formally proposed in [1]. In the independent cascade model, each edge (u, v) in the graph $G(V, E)$ is associated with a propagation probability p_{uv}, indicating the probability that u influences v after u gets influenced first. Note that, no matter u succeeds or not, it cannot attempt to influence v

---

3. It is tempting to study our problem under variable B_1 and B_2, which however makes our problem even more challenging. It is also worth noting that our current solution could provide valuable hints to the case of unfixed B_1 and B_2. Specifically, we allow an error of c when optimizing B_1 ($B_2 = B - B_1$) by requiring B_1 to take discrete values $\{c, 2c, 3c, \ldots, [B/c] \cdot c\}$. Then, the problem is reduced to the current one where B_1 and B_2 are pre-known. Thus we could derive the optimal combination of B_1 and B_2 by invoking our solutions $[B/c]$ times. Besides, we would also like to provide a heuristic method with user's degree as the proxy of its influence like [11]. To explain, we figure out the average degree of X and $N(X)$, and then determine B_1, B_2 according to the proportion of their average degrees.

TABLE 1
Frequently Used Notations

Model	
$G(V,E)$	A directed graph, where V and E represent users and the relationships respectively
$N(T)$	The neighborhood of T
X	Initially reachable users
$p_u(c_u)$	The probability that user u accepts discount c_u
$I(T)$	The expected number of users influenced by seed set T
Non-adaptive Case	
S	The set of users who becomes agents in X
C_1 (or C_2)	The discount allocation in stage 1 (or stage 2)
$P_r(S;C_1,X)$	The probability that users in S become agents if C_1 is carried out in X
$Q(C_2;N(S))$	The expected number of users influenced if C_2 is carried out in given $N(S)$
$f(C_1;X)$	The expected number of users influenced if C_1 is carried out in X
Adaptive Case	
D	The set of discounts that can be adopted
Y	The action space in stage 1
ψ	The action adopted with sequence in stage 1 ($dom(\psi)$ denotes the set of actions)
λ	The seeding realization in stage 2
ϕ	The diffusion realization
π	The adaptive seeding policy
$\hat{\sigma}(\psi,(\lambda_p,\phi_p))$	The expected number of users influenced by π under realizations λ, ϕ
$\Delta(y|\psi)$	The marginal benefit brought by y
R	Newly reachable users in stage 2 in each round
Z	The action space in stage 2 in each round

again. The influence of different edges is independent of each other. Once user v gets influenced, it will not change its state. On this basis, starting from a seed set T, the diffusion expands in discrete steps as follows. In each step, the newly influenced user, u, tries to influence its neighbors, e.g. v, along the edge (u,v), and succeeds with corresponding probability p_{uv}. The process goes on in this way, until no user is further influenced. The influence of the seed set T is denoted as $I(T)$, which is the expected number of users totally influenced, where $I(\cdot)$ is the influence function $I : 2^V \to \mathbf{R}$. As can be seen, the influence spread is closely related to the graph and propagation probabilities along edges. Thus, in computation of $I(S)$, we need full knowledge of the influence graph, i.e., $G(V,E)$ and p_{uv} ($\forall (u,v) \in E$).

To provide theoretical support for our model, by Lemma 1 we prove that the neighbors $N(X)$ have higher degree than the small set of users X in expectation. To the best of our knowledge, this is the first time that the FP phenomenon is proven to hold set-wisely in any large network. It is easy to see that the traditional point-wise FP is a special case of Lemma 1 by letting X contain only one user.

Lemma 1. *Given any connected network $G(V,E)$, consider a set of users X where each user is selected from V with probability $p \to 0$, then the friendship paradox phenomenon exists between X and $N(X)$, i.e., the average degree of X is no larger than that of $N(X)$ in expectation.*

Proof. Please refer to Section 2.1 of the supplemental file. □

4 PROBLEM FORMULATION

From the model, we can see that the influence maximization in the two-stage model consists of three sequential processes: seeding in stage 1, seeding in stage 2 and influence diffusion from stage 2. In this section, we will study two ways of seeding: the non-adaptive case and the adaptive case. In the first case, a process goes on after the previous process is finished. In the second case, the three processes iteratively go on in a circle. In each round, only part of the process is done and users are seeded based on the result of previous seeding and diffusion processes.

4.1 Non-adaptive Case

We first seed users in X to recruit agents. The discount allocation is denoted as the m-dimensional vector $C_1 = (c_1, c_2, \cdots, c_m)$, where $m = |X|$. Let S denote users who become agents in X. Then, $N(S)$ denotes friends newly reached in $N(X)$. We next allocate discounts to newly reachable users $N(S)$. Similarly, the discount allocation in $N(S)$ is denoted as $C_2 = (c_1, c_2, \cdots, c_k)$, where $k = |N(S)|$. $\forall c_i$ in C_1 or C_2, $c_i \in [0,1]$, i.e. the discounts are fractional. It is easy to see that the seed set in each stage is probabilistic, since users accept discounts with probability function $p_u(c_u)$.

Let us consider stage 1, given the allocation C_1 in X, the probability that the subset of users $S \subseteq X$ accepts the discounts is

$$P_r(S;C_1,X) = \prod_{u \in S} p_u(c_u) \prod_{v \in X \setminus S} (1 - p_v(c_v)). \quad (1)$$

Following the same technique, given allocation C_2 in $N(S)$, the probability that $T \subseteq N(S)$ becomes the seed set in stage 2 is

$$P_r(T;C_2,N(S)) = \prod_{u \in T} p_u(c_u) \prod_{v \in N(S) \setminus T} (1 - p_v(c_v)). \quad (2)$$

Then, the influence diffusion starts from the seed set T and $I(T)$ users get influenced in expectation. In stage 2, given newly reachable users $N(S)$ and the discount allocation C_2, we can obtain the expected number of users influenced, formally expressed in Equation (3),

$$Q(C_2;N(S)) = \sum_{T \subseteq N(S)} P_r(T;C_2,N(S))I(T). \quad (3)$$

Under a fixed $N(S)$, we only need to find the optimal allocation C_2 to maximize $Q(C_2;N(S))$ with budget constraint B_2. Note that $N(S)$ is probabilistically determined by the discount allocation C_1 in stage 1. To maximize the influence spread, we have to first optimize the influence spread over C_1 with stage 2 collectively considered. Given initially accessible users X, the expected influence spread with regard to C_1 is

$$f(C_1;X) = \sum_{S \subseteq X} P_r(S;C_1,X) \max Q(C_2;N(S)). \quad (4)$$

In summary, we optimize the influence spread over C_1 with possible seeding optimization in stage 2 considered. With the allocation C_1, we obtain newly reachable users $N(S)$ and further derive the optimal allocation C_2 to maximize the influence spread.

Under budget constraints B_1 and B_2, the non-adaptive influence maximization problem (NIM) can be formally formulated as follows.

NIM :

$$
\begin{aligned}
\max \quad & f(C_1; X) && \max \quad Q(C_2; N(S)) \\
\text{s.t.} \quad & \forall u \in X, c_u \in [0,1] && \text{s.t.} \quad \forall u \in N(S), c_u \in [0,1] \\
& \sum_{u \in X} c_u \leq B_1 && \qquad \sum_{u \in N(S)} c_u \leq B_2
\end{aligned}
$$

(5)

To help the reader comprehend the idea of the non-adaptive case, we provide an example in Section 1.1 of the supplemental file.

4.2 Adaptive Case

In the adaptive case, users in stage 1 are seeded sequentially, instead of computing a discount allocation of X all at once. The seeding process is defined on an action space $Y := X \times D$, where $D = \{d_1, d_2, \cdots, d_l\}$ is the set of l discrete discount rates that can be adopted. $\forall d_i \in D, d_i \in [0,1]$ and $\max\{d_i \in D\} = 1$. Selecting the action $y = (v(y), d(y)) \in Y$ means seeding user $v(y)$ with discount $d(y)$. Once $v(y)$ takes the discount, we reach its neighbors and carry out an allocation therein. Note that, the seeding processes in two stages are both based on previous diffusion results. We next introduce three basic concepts of our study.

Definition 1. (Seeding Process ψ). *ψ sequentially records the actions adopted in stage 1 which aim at reaching influential users in $N(X)$. Let $dom(\psi)$ denote the set of actions without sequence.*

Definition 2. (Seeding Realization λ). *In stage 2, for each user v in $N(X)$, v will either accept the given discount $c_v \in [0,1]$ (denoted as "1") with probability $p_v(c_v)$, or reject it (denoted as "0") with probability $1 - p_v(c_v)$. The decisions of users after being seeded in stage 2 are denoted by the function $\lambda : (N(X), [0,1]^{|N(X)|}) \to \{0,1\}$.*

Definition 3. (Diffusion Realization ϕ). *For each edge $(u, v) \in E$, it is either in "live" state (denoted as "1") or in "dead" state (denoted as "0"), indicating the influence through (u, v) is successful or not. The states of edges are denoted by the function $\phi : E \to \{0,1\}$.*

With the above preliminaries, we are ready to elaborate the sequential seeding process. Each time, we start from stage 1 and push the partial seeding process ψ_p one step forward by adopting an action $y^* = (v(y^*), d(y^*))$ from Y (i.e., probing user $v(y^*)$ with discount $d(y^*)$). If the user refuses the discount $d(y^*)$, we delete y^* from action space Y and move to the next round. Note that the budget is not wasted in this case, "refuse" means "reuse". For example, we instantiate discounts as vouchers. The discount will not be used (i.e., "reuse") if the user does not accept and apply the voucher to become a seed (i.e., "refuse"). For the case of providing samples, we can forward information of the sample to inquire users' will. If the user is not satisfied (i.e., "refuse"), we will not mail the sample to him/her and the budget is not wasted (i.e., "reuse").

On the contrary, if the user $v(y^*)$ accepts $d(y^*)$ and becomes a seed, we can reach its neighbors in stage 2. Meanwhile, actions about $v(y^*)$ are abandoned and $d(y^*)$

is subtracted from B_1. Next we allocate discounts in newly reachable users R with some budgets from B_2. Specifically, the budget drawn from B_2 only depends on the intrinsic property of $v(y^*)$, say the discount the user expects or the number of its neighbors. It is assumed that neighbors brought by each user in X are non-overlapping, since users in X are usually random. Otherwise, for a common neighbor, we can designate it as one user's neighbor. Some neighbors will accept the discounts and become seeds, forming the seed set. Then, a partial diffusion starts from the seed set and explores the state of edges exiting influenced users.

Definition 4. (Adaptive Seeding Policy π). *The adaptive policy is the function $\pi : \sigma(\psi_p, (\lambda_p, \phi_p)) \to Y$. Given observation $\sigma(\psi_p, (\lambda_p, \phi_p))$, the policy π will select an action from Y.*

After each round of seeding, ψ_p observes a partial realization (λ_p, ϕ_p). The set of users influenced under ψ_p is denoted as $\sigma(\psi_p, (\lambda_p, \phi_p))$, and the number of users influenced is $\hat{\sigma}(\psi_p, (\lambda_p, \phi_p))$. Due to the probability in users and edges, realizations λ and ϕ are both probabilistic. The prior joint probability distribution of seeding realization and diffusion realization is assumed to be $p((\lambda, \phi)) := P((\Lambda, \Phi) = (\lambda, \phi))$, where Λ is a random seeding realization and Φ is a random diffusion realization. Under realizations λ and ϕ, the seeding process determined by π is denoted as $\psi(\pi|(\lambda, \phi))$. The expected number of users influenced by π is $\hat{\sigma}(\pi) = E[\hat{\sigma}(\psi(\pi|(\Lambda, \Phi)), (\Lambda, \Phi))]$, where the expectation is calculated with respect to $p((\lambda, \phi))$. We denote the discount allocated to user $u \in X$ under π is $c(u|\pi)$. The influence maximization in the adaptive case (AIM) is a constrained optimization problem formulated as follows.

AIM :

$$
\begin{aligned}
\max \quad & \hat{\sigma}(\pi) \\
\text{s.t.} \quad & \forall u \in X, c(u|\pi) \in D, \forall(\lambda, \phi) \\
& \forall v \in N(X), c_v \in [0,1], \forall(\lambda, \phi) \\
& \sum_{u \in X} c(u|\pi) \leq B_1, \forall(\lambda, \phi) \\
& \sum_{v \in N(X)} c_v \leq B_2, \forall(\lambda, \phi)
\end{aligned}
$$

(6)

For the reader's comprehension, we provide an illustrative example in Section 1.2 of the supplemental file.

5 NON-ADAPTIVE INFLUENCE MAXIMIZATION

Based on the non-adaptive problem formulated, in this section we look into the problem and present its solution. We first analyze the properties of NIM and transform the inequality constraints in Formula (5) into equality constraints. Then, we present the two-stage coordinate descent algorithm to solve the problem.

5.1 Properties of NIM

We find that the optimization problem in NIM is NP-hard by Lemma 2. Therefore, NIM can not be solved in polynomial time unless $P = NP$.

Lemma 2. *Finding the optimal discount allocation in two stages in NIM is NP-hard.*

Lemma 3 shows that under the same budget B_2, if we reach more users in stage 2, the maximal influence spread will be larger, since we have more seeding options.

Lemma 3. *With the same budget B_2, if $T_1 \subseteq T_2$, then* $\max Q(C_2; T_1) \leq \max Q(C_2; T_2)$.

The proofs of Lemma 2 and 3 are provided in Sections 2.2 and 2.3 of the supplemental file respectively. Recall that $C_1 = (c_1, c_2, \cdots, c_m)$ and $C_2 = (c_1, c_2, \cdots, c_k)$. Let us write $C'_1 = (c'_1, \cdots, c'_m)$ and $C'_2 = (c'_1, \cdots, c'_k)$. For C_1 and C'_1 (resp. C_2 and C'_2), if $\forall i, c_i \geq c'_i$, we denote $C_1 \geq C'_1$ (resp. $C_2 \geq C'_2$). On this basis, we have the following theorem.

Theorem 1. *Monotonicity property holds in both stages, i.e.,*
 (1) If $C_2 \geq C'_2$, then $Q(C_2; N(S)) \geq Q(C'_2; N(S))$;
 (2) If $C_1 \geq C'_1$, then $f(C_1; X) \geq f(C'_1; X)$.

Proof. Please refer to Section 2.4 in the supplemental file. □

Theorem 1 indicates that the more discount we allocate to users in each stage, the larger the influence spread is. Thus, we can draw the conclusion that the budget allocated to both stages will be used up. By contradiction, if there is remaining budget in stage 1 (resp. stage 2) while f (resp. Q) is maximized, we can add it to current discount allocation C_1 (resp. C_2). Then, f (resp. Q) is further increased, a contradiction. Thus, the NIM is equivalent to the following problem.

$$
\begin{aligned}
&\max \quad f(C_1; X) &&\max \quad Q(C_2; N(S)) \\
&\text{s.t.} \quad \forall u \in X, c_u \in [0,1] &&\text{s.t.} \quad \forall u \in N(S), c_u \in [0,1] \\
&\qquad \sum_{u \in X} c_u = B_1 &&\qquad \sum_{u \in N(S)} c_u = B_2
\end{aligned} \tag{7}
$$

5.2 Coordinate Descent Allocation

We attempt to decide the discount allocation in each stage by the coordinate descent algorithm. Since the design of C_1 should collectively consider the allocation in stage 2, for readability, we first explain how to decide the allocation in stage 2. Following a similar idea, we design the allocation in stage 1.

5.2.1 Coordinate Descent in Stage 2

Given the seed set S in stage 1 and the budget constraint B_2, the coordinate descent algorithm iteratively optimizes $Q(C_2; N(S))$ from an initial allocation in $N(S)$, e.g., uniform allocation. In each iteration, we randomly pick two users u and v, whose discounts are c_u and c_v respectively. Then, we adjust the discounts between u and v to optimize $Q(C_2; N(S))$, with other users' discounts fixed.

Let $B'_2 = c_u + c_v$, B'_2 is a constant during the rearrangement. Similar to the expansion of the objective function in [8], we could rewrite $Q(C_2; N(S))$ w.r.t. c_u as follows

$$
\begin{aligned}
Q(C_2; N(S)) = &\sum_{T \subseteq N(S) \backslash \{u,v\}} P_r(T; C_2, N(S) \backslash \{u,v\}) \cdot \\
&\Big\{ [1 - p_u(c_u)][1 - p_v(B'_2 - c_u)] I(T) \\
&+ [1 - p_u(c_u)] p_v(B'_2 - c_u) I(T \cup \{v\}) \\
&+ p_u(c_u)[1 - p_v(B'_2 - c_u)] I(T \cup \{u\}) \\
&+ p_u(c_u) p_v(B'_2 - c_u) I(T \cup \{u,v\}) \Big\}.
\end{aligned} \tag{8}
$$

Thus, $Q(C_2; N(S))$ is a function w.r.t. c_u. Then, we can write it as $Q(c_u)$. Due to $0 \leq c_u, c_v \leq 1$ and $c_u + c_v = B'_2$, we have the constraint $\max(0, B'_2 - 1) \leq c_u \leq \min(B'_2, 1)$.

In each iteration, we obtain the new discount of u by solving the following optimization problem. In the mean time, c_v is determined by $c_v = B'_2 - c_u$.

$$
\begin{aligned}
&\max \quad Q(c_u) \\
&\text{s.t.} \quad \max(0, B'_2 - 1) \leq c_i \leq \min(B'_2, 1)
\end{aligned} \tag{9}
$$

It is an optimization over a single variable in a closed interval. Since $p_u(\cdot)$ and $p_v(\cdot)$ are continuously differentiable, $Q(c_u)$ is differentiable as well. Therefore, the discount c_u that maximizes $Q(c_u)$ must be in one of the three cases: (1) the stationary points of $Q(c_u)$ in interval $(\max(0, B'_2 - 1), \min(B'_2, 1))$; (2) $\max(0, B'_2 - 1)$; (3) $\min(B'_2, 1)$. We check the value of $Q(c_u)$ in the three cases and choose the optimal one as the new discount allocated to user u.

Algorithm 1 The Coordinate Descent Algorithm CD(Q,B,T)

Input: Objective function $Q(\cdot)$, budget B, accessible users T
Output: Allocation C
1: Initialize C
2: **while** not converge **do**
3:     Randomly pick users u and $v \in T$
4:     $B' \leftarrow c_u + c_v$
5:     Find all stationary points x of $Q(c_u)$ in $(\max(0, B' - 1), \min(B', 1))$
6:     $c_u \leftarrow \operatorname{argmax}_{\{x, \max(0, B'-1), \min(B', 1)\}} Q(c_u)$
7:     $c_v \leftarrow B' - c_u$
8: **end while**
9: Return C

The coordinate descent algorithm is described in Algorithm 1. Its convergence is guaranteed, since $Q(c_u)$ is upper bounded by the size of the network, and $Q(c_u)$ is monotonically increasing as the iteration goes on. Thus, the algorithm will finally converge to a limit allocation. However, we can not ensure the limit to be locally optimal, which depends on the property of $Q(c_u)$. Even if the limit is locally optimal, it may not be the global optimum, where different initial allocations could be applied for improvement.

5.2.2 Coordinate Descent in Stage 1

Considering the hardness of computing the global optimum of $Q(C_2; \cdot)$, which is needed in $f(C_1; X)$ by Eq. 4, we would like to define a proxy function $\hat{Q}(C_2; \cdot)$ which is maximized by Alg. 1. Accordingly, the objective function becomes $\hat{f}(C_1; X)$. The coordinate descent allocation in stage 1 follows the similar idea. We start from an arbitrary allocation in X and optimize $\hat{f}(C_1; X)$ iteratively. In each iteration, we randomly pick two users i, j and adjust the discounts between them with other discounts fixed. Let $B'_1 = c_i + c_j$, B'_1 is a constant during the rearrangement.

Then, $\hat{f}(C_1; X)$ can be written as

$$
\begin{aligned}
\hat{f}(C_1; X) = \sum_{S \subseteq X \setminus \{i,j\}} & P_r(S; C_1, X \setminus \{i,j\}) \cdot \\
& \Big\{ [1 - p_i(c_i)][1 - p_j(B_1' - c_i)] \max \hat{Q}(C_2; N(S)) \\
& + [1 - p_i(c_i)]p_j(B_1' - c_i) \max \hat{Q}(C_2; N(S \cup \{j\})) \\
& + p_i(c_i)[1 - p_j(B_1' - c_i)] \max \hat{Q}(C_2; N(S \cup \{i\})) \\
& + p_i(c_i)p_j(B_1' - c_i) \max \hat{Q}(C_2; N(S \cup \{i,j\})) \Big\}.
\end{aligned}
\tag{10}
$$

To handle $\hat{f}(C_1; X)$, we need to optimize $\hat{Q}(C_2; \cdot)$ and estimate its influence spread. Note that the optimal allocation of $\hat{Q}(C_2; \cdot)$ could be obtained by Alg. 1. To estimate the resultant influence spread, we apply a polling based method in Section 3 of the supplementary file. Analogous to the analysis in stage 2, $\hat{f}(C_1; X)$ can be written as $\hat{f}(c_i)$. In each iteration, we obtain the new discount of i by solving the following optimization problem in the same way as stage 2. Meanwhile, c_j is determined according to $c_j = B_1' - c_i$.

$$
\begin{aligned}
\max \quad & \hat{f}(c_i) \\
\text{s.t.} \quad & \max(0, B_1' - 1) \leq c_i \leq \min(B_1', 1)
\end{aligned}
\tag{11}
$$

A framework of deciding discounts with coordinate descent algorithm is established in Algorithm 2. We first design the allocation in stage 1 with stage 2 collectively considered. Then, based on the seeding result in stage 1, we determine the allocation in stage 2.

Algorithm 2 The Two-Stage Coordinate Descent Framework

Input: Budget B_1, B_2, initially reachable users X
Output: Allocation C_1, C_2
1: $C_1 \leftarrow CD(\hat{f}(C_1; X), B_1, X)$
2: Seed users in X with C_1
3: $S \leftarrow$ users in X who accept discounts
4: $C_2 \leftarrow CD(\hat{Q}(C_2; N(S)), B_2, N(S))$
5: Return C_1, C_2

---

6 ADAPTIVE INFLUENCE MAXIMIZATION

In the adaptive case, the seeding and diffusion processes iteratively go on in a circle. In each round, one user is seeded in stage 1. Discounts are allocated to the newly reached neighbors in stage 2, and then the influence spread expands. The adaptive case is studied in two discount settings: discrete-continuous setting and discrete-discrete setting.

6.1 Discrete-Continuous Setting

Discrete-Continuous Setting: Users in stage 1 are probed with actions from $Y = X \times D$, where $D = \{d_1, d_2, \cdots, d_l\}$ is the set of optional discount rates (*discrete*). In stage 2, discounts of newly reachable users take value in interval $[0, 1]$ (*continuous*).

6.1.1 Seeding Strategy

In this subsection, we first specify the selection of actions in stage 1. Given the previous seeding process ψ_p in stage 1, recall that $\sigma(\psi_p, (\lambda_p, \phi_p))$ denotes the set of influenced users under ψ_p. Without causing ambiguity, we will write

$\sigma(\psi_p) = \sigma(\psi_p, (\lambda_p, \phi_p))$. Then, the induced graph of uninfluenced users $V \setminus \sigma(\psi_p)$ can be denoted as $G(V \setminus \sigma(\psi_p))$. Let $\Delta(y|\psi_p)$ denote the expected number of users influenced in $G(V \setminus \sigma(\psi_p))$, if $v(y)$ becomes the seed and discounts are allocated to $v(y)$'s newly reached neighbors. $\Delta(y|\psi)$ is the marginal benefit brought by y, expressed as

$$
\begin{aligned}
\Delta(y|\psi_p) := E[\hat{\sigma}(\psi_p \cup \{y\}, (\Lambda, \Phi)) - \\
\hat{\sigma}(\psi_p, (\Lambda, \Phi))|(\Lambda, \Phi) \sim \psi_p],
\end{aligned}
\tag{12}
$$

where $(\Lambda, \Phi) \sim \psi_p$ denotes random realizations that contain the existing realization observed by ψ_p, and the expectation is taken with respect to $p((\lambda, \phi)) := P((\Lambda, \Phi) = (\lambda, \phi))$. In each round, we select the action that maximizes the benefit-to-cost ratio with the remaining budget, i.e.

$$
y^* = \operatorname*{argmax}_{y \in Y} \frac{\Delta(y|\psi_p)}{d(y)}.
\tag{13}
$$

Supposing the targeted user $v(y^*)$ accepts the discount, we reach its neighbors and come to stage 2, where we further allocate discounts to newly reached neighbors R according to the coordinate descent algorithm $CD(\cdot)$. However, if the user refuses the discount, we remove y^* from Y and proceed to the next round. Based on the above description, the pseudo-code is presented in Algorithm 3 π^{greedy}. The $DCA(\cdot)$ function in Algorithm 3 is presented in Algorithm 4, which describes the actions triggered after action y^* is accepted. The framework of π^{greedy} will be inherited in the subsequent adaptive greedy algorithms, except some modifications.

Algorithm 3 The Adaptive Greedy Algorithm π^{greedy}

Input: Budget B_1, B_2, action space $Y = X \times D$
Output: Accepted actions P_1, allocation C_2
1: Initialize $P_1 \leftarrow \emptyset$, $C_2 \leftarrow \mathbf{0}$
2: **while** $B_1 \geq 0$ **do**
3:     **if** $\exists y \in Y$ s.t. $d(y) \leq B_1$ **then**
4:         Select $y^* = \operatorname{argmax}_{y \in Y} \frac{\Delta(y|\psi_p)}{d(y)}$ s.t. $d(y) \leq B_1$,
5:         Probe $v(y^*)$ with discount $d(y^*)$
6:         **if** $v(y^*)$ accepts $d(y^*)$ **then**
7:             $R \leftarrow$ newly reachable users
8:             $C_2^p \leftarrow DCA(R, y^*, P_1, B_1, Y)$
9:             Update C_2 with C_2^p
10:         **else**
11:             $Y \leftarrow Y \setminus y^*$
12:         **end if**
13:     **end if**
14: **end while**
15: Return P_1, C_2

Algorithm 4 D-C Allocation $DCA(R, y^*, P_1, B_1, Y)$

Input: Newly reachable users R, action newly accepted y^*, accepted actions P_1, remaining budget B_1, current action space Y
Output: Allocation C_2^p
1: Initialize C_2^p
2: $P_1 \leftarrow P_1 \cup y^*$; $B_1 \leftarrow B_1 - d(y^*)$
3: $Y \leftarrow Y \setminus \{y|v(y) = v(y^*)\}$
4: $C_2^p \leftarrow CD(\hat{Q}(C_2^p; R), \frac{B_2}{B - B_2} \times d(y^*), R)$
5: Return C_2^p

Definition 5. (Adaptive Submodularity). *The function $\hat{\sigma}$ is adaptive submodular with respect to $p((\lambda, \phi))$, if for all $\psi \subseteq \psi'$(i.e.ψ is a subprocess of ψ'), and $\forall y \in Y \setminus dom(\psi)$, we have*

$$\Delta(y|\psi) \geq \Delta(y|\psi').$$

Furthermore, if for all ψ and all $y \in Y$, $\Delta(y|\psi) \geq 0$ holds, then $\hat{\sigma}$ is adaptive monotone.

By Lemma 4, we find that $\hat{\sigma}(\cdot|(\lambda, \phi))$ is adaptive submodular and adaptive monotone, where the proof is deferred to Section 2.5 of the supplemental file. Since a non-negative linear combination of monotone adaptive submodular functions is still monotone adaptive submodular, we have $\hat{\sigma}(\cdot)$ is monotone and adaptive submodular.

Lemma 4. $\hat{\sigma}(\cdot|(\lambda, \phi))$ *is adaptive monotone and adaptive submodular, under any realization (λ, ϕ).*

6.1.2 Relaxation Analysis

Following a similar idea of analyzing the adaptive algorithm in [9], we relax the seeding process in stage 1 by assuming that the minimum discount rate $d_{\min}(u) \in D$ desired by each user u is pre-known. In the seeding process, if u is probed with a discount no smaller than $d_{\min}(u)$, then u will accept it and become a seed, and vice versa. To maximize the benefit-to-cost ratio, each user will be probed from small discounts. By the definition of $d_{\min}(u)$, discounts smaller than $d_{\min}(u)$ will be rejected. When the discount becomes $d_{\min}(u)$, the user will accept it and become the seed, and polices with higher discount are abandoned. Thus, it becomes meaningless to probe user u with discounts higher or lower than $d_{\min}(u)$. And, the action space is reduced to $Y^{\text{relaxed}} = \{(u, d_{\min}(u)), u \in X\}$. We denote the adaptive greedy algorithm under the relaxed setting as $\pi_{\text{relaxed}}^{\text{greedy}}$. In each round, we select an action y^* from Y^{relaxed} that maximizes the benefit-to-cost ratio $\frac{\Delta(y|\psi_p)}{d_{\min}(v(y))}$. Once an action in Y^{relaxed} is adopted, the user will become a seed definitely, since $d_{\min}(u)$ is the desired discount of user u. The subsequent seeding strategy in newly reachable users remains unchanged.

With the relaxation analysis, we find that the seed sets of the relaxed setting and the original setting are the same in stage 1 by Lemma 5, which is similar to [9]. This lemma is proven by mathematical induction in Section 2.6 of the supplemental file.

Lemma 5. *Under any realization (λ, ϕ), $\pi_{\text{relaxed}}^{\text{greedy}}$ yields the same seed set in stage 1 as π^{greedy}.*

Let π^{OPT} denote the optimal policy under the discrete-continuous setting. With the adaptive submodularity of $\hat{\sigma}(\cdot)$ and Lemma 5, we obtain the performance guarantee of the greedy algorithm π^{greedy} in Theorem 2.

Theorem 2. *If global optimality is obtained in stage 2 in each round, then the adaptive greedy policy π^{greedy} obtains at least $(1 - e^{-\frac{B_1-1}{B_1}})$ of the value of the optimal policy π^{OPT},*
$$\hat{\sigma}(\pi^{\text{greedy}}) \geq (1 - e^{-\frac{B_1-1}{B_1}})\hat{\sigma}(\pi^{\text{OPT}}).$$

Proof. Please refer to Section 2.7 in the supplemental file.
□

6.2 Discrete-Discrete Setting

In the previous setting, the coordinate descent algorithm is applied to decide the continuous allocation in stage 2, which needs numerous iterative optimizations. In this subsection, we introduce a discrete solution in stage 2.

Discrete-Discrete Setting: In each stage, we select actions from an action space defined by the Cartesian product of users and discount rates D. Thus, the discounts in both stages are discrete.

Definition 6. (Submodularity) *For a real-valued function $h(\cdot)$ defined on subsets of a finite ground set G. If for all $A \subseteq B \subseteq G$, and for all $x \in G \setminus B$, we have*
$$h(A \cup \{x\}) - h(A) \geq h(B \cup \{x\}) - h(B).$$

Then, we say $h(\cdot)$ is submodular. Furthermore, if $h(A) \leq h(B)$ holds for all $A \subseteq B \subseteq G$, $h(\cdot)$ is said to be monotone.

The seeding process in stage 1 is still sequential. In each round, we select an action y^* from $Y := X \times D$. If $v(y^*)$ refuses the discount $d(y^*)$, we delete y^* from Y and move to the next round. If $v(y^*)$ accepts it and becomes a seed, some users become newly reachable, denoted as R. An action space $Z = R \times D$ is defined in stage 2. We will select a subset of actions $L \subseteq Z$ to seed users in R under some budgets drawn from B_2. The budget only depends on the intrinsic property of $v(y^*)$, just like the discrete-continuous setting. Different from selecting actions sequentially in stage 1, we decide actions in stage 2 all at once without observing the seeding and diffusion results of each action. Because in realistic scenario, we are not likely to have so much time to observe the diffusion of each action in stage 2.

We next show how to select the set of actions in stage 2. Let $d(u|L)$ denote the discount allocated to user u under actions L. Analogous to the definition in Equations (1) and (3), we define the probability that a subset of users $T \subseteq R$ accept discounts as
$$P_r(T; L, R) = \prod_{u \in R} p_u(d(u|L)) \prod_{v \in R \setminus S} (1 - p_v(d(v|L))). \quad (14)$$

Given the partial seeding process ψ_p in stage 1 and the set of influenced users $\sigma(\psi_p)$, we denote the expected number of users newly influenced by T as $I_{G(V \setminus \sigma(\psi_p))}(T)$. Then, the number of users influenced by actions L is
$$Q(L; R) = \sum_{T \subseteq R} P_r(T; L, R) I_{G(V \setminus \sigma(\psi_p))}(T). \quad (15)$$

We attempt to find a set of actions that maximize the influence spread. However, we find that maximizing $Q(L; R)$ is NP-hard by Lemma 6.

Lemma 6. *Finding the optimal set of actions in stage 2 is NP-hard.*

However, we can prove the monotonicity and submodularity of $Q(L; R)$ by Lemma 7. Since $Q(L; R)$ is monotone and submodular, we are motivated to design the approximate algorithm $GS(R, y^*, P_1, B_1, Y)$ in Algorithm 5 to determine the actions in stage 2.

Lemma 7. $Q(L; R)$ *is monotone and submodular w.r.t. L.*

The selection of action y^* in stage 1 is similar to the discrete-continuous setting. We select the one that maximizes the benefit-to-cost ratio $\frac{\Delta(y|\psi_p)}{d(y)}$. Note that, when calculating $\Delta(y|\psi_p)$, discounts of users in stage 2 are de-

termined by Algorithm 5 rather than the coordinate descent algorithm.

We continue to examine the property of $\hat{\sigma}(\cdot|(\lambda, \phi))$ in the discrete-discrete setting and find that it is still adaptive submodular by Lemma 8.

Lemma 8. *In the Discrete-Discrete Setting, $\hat{\sigma}(\cdot|(\lambda, \phi))$ is still adaptive submodular, under any realization (λ, ϕ).*

For readability, the proofs of Lemma 6, 7 and 8 in this subsection are deferred to Sections 2.8, 2.9 and 2.10 of the supplemental file respectively. Based on the above description, we obtain the adaptive greedy algorithm $\pi_{\text{discrete}}^{\text{greedy}}$ in the discrete-discrete setting. Since the only difference from π^{greedy} lies in the allocation in stage 2, we can derive $\pi_{\text{discrete}}^{\text{greedy}}$ by replacing $DCA(R, y^*, P_1, B_1, Y)$ with $GS(R, y^*, P_1, B_1, Y)$. For the sake of the space, we omit the detailed description here. Let $\pi_{\text{discrete}}^{\text{OPT}}$ denote the optimal policy. With the adaptive submodularity in stage 1 and submodularity in stage 2, we obtain the performance guarantee of $\pi_{\text{discrete}}^{\text{greedy}}$ in Theorem 3.

Algorithm 5 The Greedy Selection $GS(R, y^*, P_1, B_1, Y)$

Input: Newly reachable users R, action newly accepted y^*, actions accepted P_1, remaining budget B_1, current action space Y
Output: Actions P_2
1: Initialize $P_2 \leftarrow \emptyset$
2: $P_1 \leftarrow P_1 \cup \{y^*\}$; $B_1 \leftarrow B_1 - d(y^*)$
3: $Y \leftarrow Y \setminus \{y|v(y) = v(y^*)\}$
4: $S_1 \leftarrow \emptyset, Z \leftarrow R \times D, S_2 \leftarrow \text{argmax}_{z \in Z}\{Q(\{z\}; R)|z \in Z, d(z) \leq \frac{B_2}{B-B_2} \times d(y^*)\}$
5: **while** $\sum_{z \in S_1} d(z) \leq \frac{B_2}{B-B_2} \times d(y^*)$ **do**
6:     $z^* \leftarrow \text{argmax}_{z \in Z} \frac{Q(S_1 \cup z; R) - Q(S_1)}{d(z)}$
7:     **if** $d(z^*) + \sum_{z \in S_1} d(z) \leq \frac{B_2}{B-B_2} d(y^*)$ **then**
8:         $S_1 \leftarrow S_1 \cup \{z^*\}, Z \leftarrow Z \setminus \{z^*\}$
9:     **end if**
10: **end while**
11: $P_2 \leftarrow \text{argmax}_{S \in \{S_1, S_2\}} Q(S; R)$
12: Return P_2

Theorem 3. *The greedy policy $\pi_{\text{discrete}}^{\text{greedy}}$ obtains at least $(1 - e^{-\frac{B_1-1}{2B_1}(1-\frac{1}{e})})$ of the value of $\pi_{\text{discrete}}^{\text{OPT}}$,*

$$\hat{\sigma}(\pi_{\text{discrete}}^{\text{greedy}}) \geq (1 - e^{-\frac{B_1-1}{2B_1}(1-\frac{1}{e})})\hat{\sigma}(\pi_{\text{discrete}}^{\text{OPT}}).$$

Proof. Please refer to Section 2.11 in the supplemental file. □

The performance guarantee of $\pi_{\text{discrete}}^{\text{greedy}}$ is not appealing enough. The enumeration method can be further applied to improve the approximation ratio. In fact, the size of newly reachable users is relatively small compared with the whole network, hence, the enumeration will not be so computationally costly. The modified greedy algorithm in stage 2 is described in Algorithm 6 whose approximation ratio is $(1-e^{-1})$ [29]. The complete algorithm $\pi_{\text{enum}}^{\text{greedy}}$ can be derived by replacing $DCA(R, y^*, P_1, B_1, Y)$ in Algorithm 3 with the $MGS(R, y^*, P_1, B_1, Y)$. Following similar argument in the proof of Theorem 3, we obtain the approximation ratio of $\pi_{\text{enum}}^{\text{greedy}}$ in Theorem 4.

Theorem 4. *The policy $\pi_{\text{enum}}^{\text{greedy}}$ achieves an approximation ratio of $1 - e^{-\frac{B_1-1}{B_1}(1-\frac{1}{e})}$.*

Algorithm 6 Modified Greedy $MGS(R, y^*, P_1, B_1, Y)$

Input: Newly reachable users R, action newly accepted y^*, actions accepted P_1, remaining budget B_1, current action space Y
Output: Actions P_2
1: $P_1 \leftarrow P_1 \cup \{y^*\}$; $B_1 \leftarrow B_1 - d(y^*)$
2: $Y \leftarrow Y \setminus \{y|v(y) = v(y^*)\}$
3: $S_1 \leftarrow \emptyset, Z \leftarrow R \times D, S_2 \leftarrow \text{argmax}\{Q(A; R)|A \subseteq Z, |A| < 3, \sum_{z \in A} d(z) \leq \frac{B_2}{B-B_2} d(y^*)\}$
4: **for** $A \subseteq Z, |A| = 3, \sum_{z \in A} d(z) \leq \frac{B_2}{B-B_2} d(y^*)$ **do**
5:     $Z' \leftarrow Z \setminus A$
6:     **while** $Z' \neq \emptyset$ **do**
7:         $z^* \leftarrow \text{argmax}_{z \in Z'} \frac{Q(A \cup z; R) - Q(A; R)}{d(z)}$
8:         **if** $d(z^*) + \sum_{z \in A} d(z) \leq \frac{B_2}{B-B_2} d(y^*)$ **then**
9:             $A \leftarrow A \cup \{z^*\}$
10:         **end if**
11:         $Z' \leftarrow Z' \setminus z^*$
12:     **end while**
13:     **if** $Q(A) > Q(S_1)$ **then**
14:         $S_1 \leftarrow A$
15:     **end if**
16: **end for**
17: $P_2 \leftarrow \text{argmax}_{S \in \{S_1, S_2\}} Q(S; R)$
18: Return P_2

---

7 EXPERIMENTS

In this section, we examine the performance of the proposed algorithms on four real-world datasets. The purpose lies in three parts. First, we compare the expected influence spread of our two-stage algorithms with other algorithms to show the advantage of our proposed algorithms. Second, we examine the scalability of our algorithms with respect to the total budget. Third, we test the sensitivity to different settings of seed probability function. Fourth, the impact of the friendship paradox is evaluated. All algorithms were implemented in C# and simulated on a Linux x64 server (Intel Xeon E5-2650 v2 @2.6Ghz, 128GB RAM).

7.1 Experimental setup

Dataset Description. We test our algorithms on four networks derived from SNAP [30]. The parameters of the four datasets are presented in Table 2. Undirected networks are converted to directed networks, which means that every undirected edge (u, v) is replaced by two directed edges (u, v) and (v, u), and the number of edges is doubled. In each network, we randomly selected 100 nodes as the initially accessible users X.

TABLE 2
Datasets

Datasets	Nodes	Edges	Type	θ
wiki-Vote	7,115	103,689	Directed	0.25M
ca-CondMat	23,133	186,936	Directed	2M
com-DBLP	317,080	1,049,866	Undirected	20M
soc-LiveJournal1	4,847,571	68,993,773	Directed	40M

Propagation Probability. The diffusion model adopted is the independent cascade model, which is widely employed

in the literature of influence maximization [7, 8, 21, 31]. Each edge (u, v) is associated with a propagation probability p_{uv}, set to be $\frac{\alpha}{\text{in-degree of } v}$, where $\alpha \in \{0.6, 0.8, 1\}$. This setting is quite common in existing works [1, 8, 9, 31].

Seed Probability Function. Recall that whether a user u accepts the discount c_u is captured by the seed probability function $p_u(c_u)$, which means the probability that u accepts the discount c_u and becomes a seed. User's behavior is affected by various factors, such as time and demand. The best way to estimate the probability function may be learning from data, which is out of the scope of our research. Thus, we apply synthesized seed probability functions, which satisfy the four properties mentioned in Section 3. For each network, we randomly select 5% nodes and set $p_u(c_u)$ with $p_u(c_u) = c_u^2$, 10% nodes with $p_u(c_u) = c_u$ and 85% with $p_u(c_u) = 2c_u - c_u^2$.

Discount Rate. The action space is defined as the Cartesian product of users and discount rates D. In our experiment, the discount rate D is set to be an arithmetic progression from 10% to 100% with common difference 10%. That is, 10 candidate discounts are considered.

Implementation. Influence estimation is frequently demanded in the algorithms to determine discount allocation. To obtain an unbiased estimation, we adopt the polling based technique proposed in [7] [21], where θ (given in Table 2) reverse reachable sets are generated. For detailed description, please refer to Section 3 of the supplemental file. All the reported influence spreads are estimated by running 20K times Monte Carlo simulations.

7.2 Algorithms Evaluated

To validate the performance of our four algorithms, we include six more algorithms for comparison. All the algorithms are tested under budgets $B \in \{10, 20, 30, 40, 50\}$ with $B_1 : B_2 = 1 : 4$. For one-stage algorithms, the budget B is all spent in X.

- **Non-adaptive Algorithms**

  Random Friend (RF): We introduce RF as a basic two-stage algorithm. We uniformly and randomly select B_1 users in X as agents S. Then, B_2 users are selected from $N(S)$ in a similar way.

  Discrete Influence Maximization (IM): First applied by Kempe *et al.* [1], IM is a classic algorithm with performance guarantee $1 - \frac{1}{e}$. Users in X are greedily selected in a manner that, given the first k seeds, the user u, which maximizes the expected marginal benefit $I(T \cup \{u\}) - I(T)$, is selected as the $(k + 1)$-th seed.

  Coordinate Descent (CD): This is a one-stage algorithm carried out in initially accessible users X. Detailed description could be found in Algorithm 1. As for the initial allocation, we first rank users in X with respect to the degree in a non-increasing order. Then, the budget B is uniformly allocated to the first $1.5B$ users. The number of iterations is set to be 50, enough for the refinement in 100 nodes.

  Two-stage CD (2CD): This is the two-stage algorithm described in Algorithm 2. The initial allocation in each stage is determined in the same way as CD. The number of iterations is 10 in both stages.

- **Adaptive Algorithms**

Adaptive Selection (Ada): As the only one-stage algorithm in the adaptive case, adapted from [9], Ada sequentially seeds users in X with budget B. Each time, we select the action y from Y which maximizes the benefit-to-cost ratio $\frac{\Delta(y|\psi_p)}{d(y)}$, where $\Delta(y|\psi_p)$ is the expected influence spread brought by $v(y)$ under the previous diffusion result.

LP-Based Approach (LP): A two-stage algorithm proposed in [11], where the degree of a user is regarded as its influence. The maximization problem is formulated as an integer linear programming. The solution returns an allocation with $(1 - \frac{1}{e})$-approximation ratio.

A-Greedy: Proposed in [32], A-Greedy is an adaptive one-stage algorithm. Thus, we transform it into a two-stage algorithm. In both stages, A-Greedy is applied to select the seeds. The optimal seeding pattern A^* is adopted, where one user is selected each time and the next selection takes place until no user is further influenced. The activation probability f_u of each user u in [32] is set to be 0.6.

π^{greedy} **(Ada+CD):** This two-stage algorithm is described in detail in Alg. 3. The coordinate descent algorithm is applied in newly reachable users. The initialization and number of iterations are the same as the CD algorithm.

$\pi^{\text{greedy}}_{\text{discrete}}$ **(Ada+GS):** The framework is the same as the Ada+CD algorithm. But in stage 2, the coordinate descent algorithm is replaced by the Greedy Selection in Alg. 5. It is worth noting that the discount rate D in stage 2 becomes $\{0.5, 1\}$, because we find that a fine-grained discount rate with granularity 0.1 will lead to worse results. The explanation is that greedy selection prefers giving small discounts to many users, while the number of newly reachable users is relatively small. Then, the budget is left, making the experiment unfair.

$\pi^{\text{greedy}}_{\text{enum}}$ **(Ada+MGS):** The actions in the second stage are determined by the Modified Greedy Selection described in Alg. 6. The discount rate in stage 2 remains to be $\{0.5, 1\}$ for the same reason.

7.3 Experimental Results

The ten algorithms are mainly evaluated on four metrics: (1) the expected influence spread; (2) the scalability regarding the total budget; (3) the sensitivity of algorithms with respect to different settings of seed probability functions; (4) the impact of the FP phenomenon. Due to space limitation, the influence spread and scalability under $\alpha = 0.8$ are presented in Section 4 of the supplemental file.

7.3.1 Influence Spread

The expected influence spread of the non-adaptive case and the adaptive case is presented in Fig. 2 and Fig. 3 respectively.

- **Non-adaptive Case**

  From Fig. 2, we can see that CD performs better than IM, since the discount allocation in CD is allowed to be fractional and thus more fine-grained. It is a little surprising that the simple two-stage algorithm RF shows larger influence spread than the elaborate CD in most settings, except two points in Wiki-Vote ($\alpha = 0.6$, $B = 10$ and 30) due to its randomness. The reason is that RF has access to influential neighbors. As can be seen, the two-stage coordinate descent algorithm outperforms the other three algorithms in all the
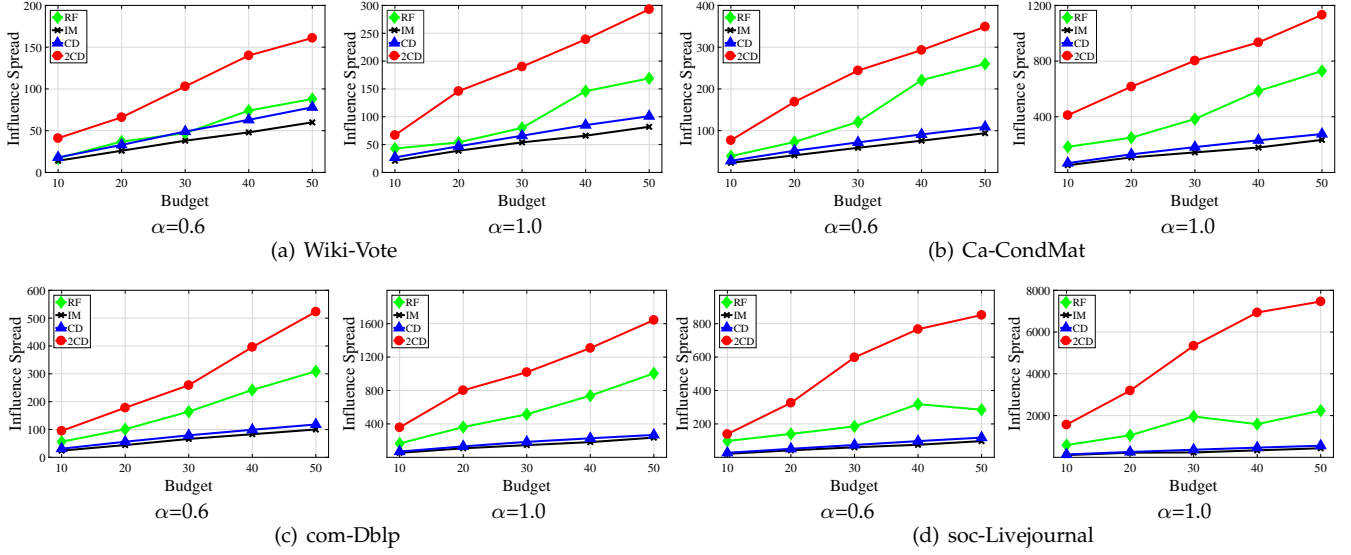
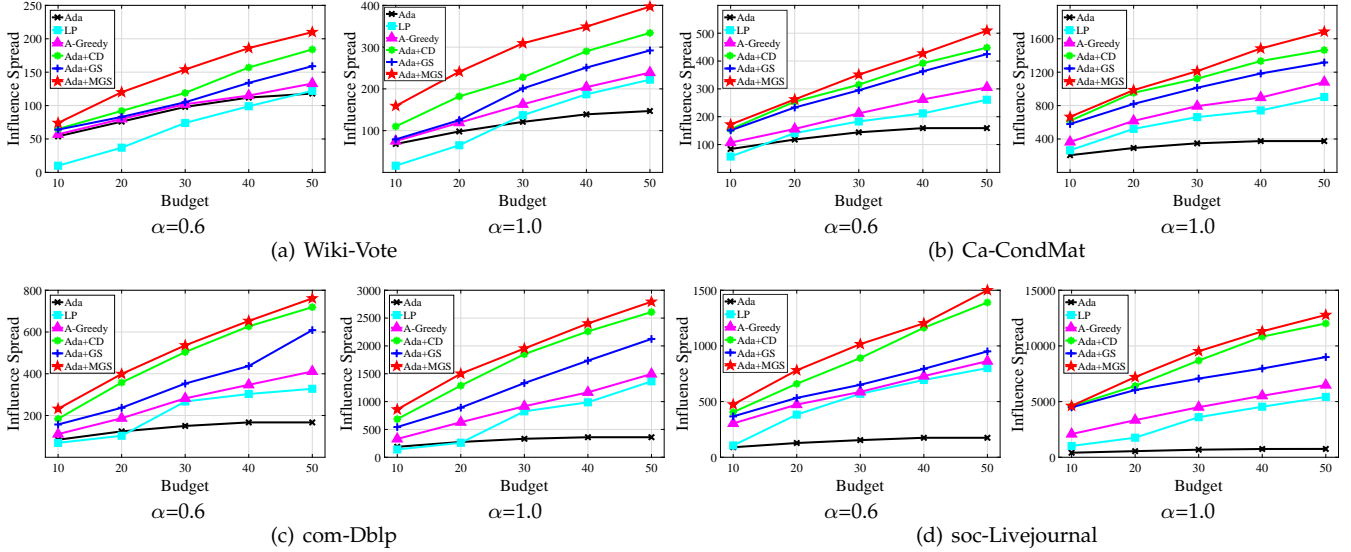Fig. 2. Influence Spread in the Non-adaptive Case.

Fig. 3. Influence Spread in the Adaptive Case.

settings. The ratio between 2CD and the second best result varies from 1.3 to nearly 3. This result is easy to understand since 2CD not only has access to the influential neighbors but also makes refinements in both stages.

Moreover, although the scale of the four datasets is quite different, we find that the influence spread of one-stage algorithms (i.e., CD and IM) is nearly in the same scale, while the influence spread of two-stage algorithms scales as the size of networks. We can infer that simply allocating discounts to initially accessible users restricts the spread of influence. Meanwhile, exploiting the friendship paradox helps expand the influence spread.

• **Adaptive Case**

As can be seen from Fig. 3, in most settings the one-stage adaptive algorithm Ada has the smallest influence spread, since only initially accessible users are seeded. In some settings, the LP algorithm is even worse than Ada, especially in small datasets and small budgets, since in LP the degree is directly regarded as the influence. This treatment helps with the complexity but losses accuracy,

and causes possible blindness when selecting seeds. In A-Greedy, the influence is unbiasedly estimated by the hypergraph. The result of A-Greedy is thus better than that of LP. Our proposed three algorithms achieve larger influence spread than the above three algorithms. The reason is three-fold. First, the discount is fractional and thus more fine-grained. Second, the influence is accurately estimated. Third, the FP phenomenon is leveraged. There is an evident gap between Ada+MGS and Ada+GS. This phenomenon conforms with the theoretical result that the approximation ratio of Ada+MGS is larger than Ada+GS. In the experiment, Ada+CD shows smaller influence spread than Ada+GS and Ada+MGS. However, we can not say that Ada+CD is definitely inferior to the other two algorithms, since its performance is closely related to the initial allocation. With a better initial allocation and more iterations, Ada+CD could deliver a better performance.

Comparing our four proposed algorithms, we can see that adaptive algorithms yield larger influence spread than 2CD, except some minor points in the Wiki-Vote dataset.
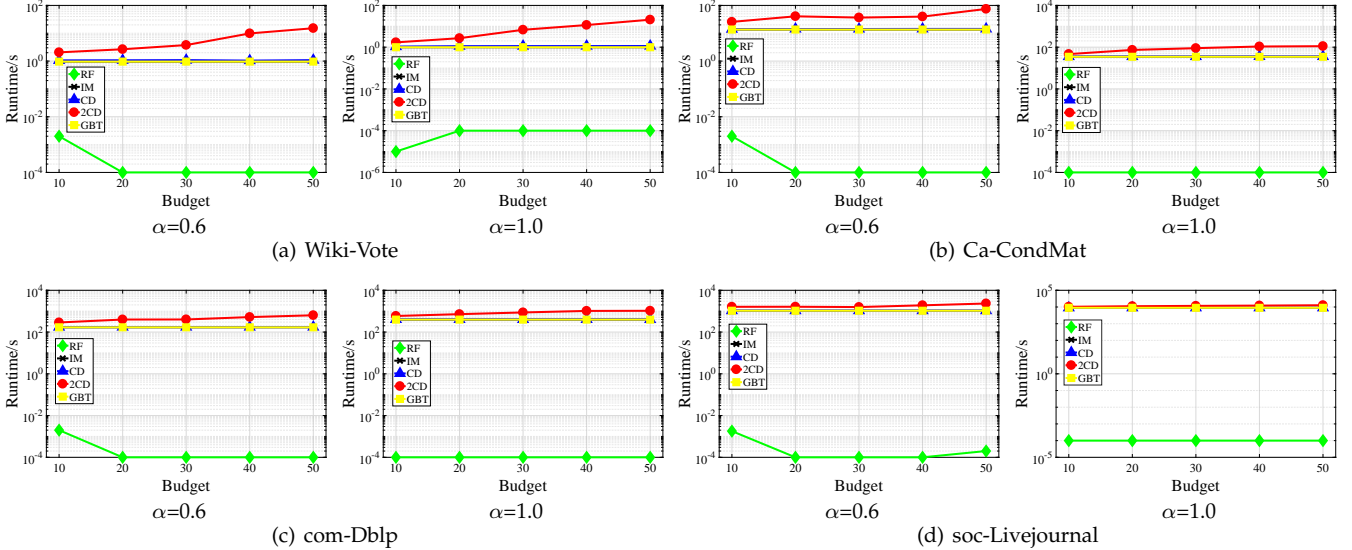
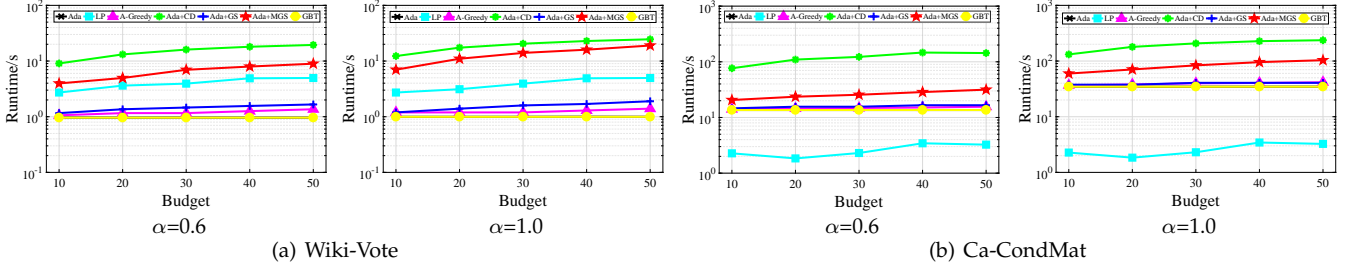Fig. 4. Running Time in the Non-adaptive Case.

Fig. 5. Running Time in the Adaptive Case.

The reason is that adaptive algorithms make the most of the remaining budget by seeding the next user wisely based on the observation on the previous influence spread.

7.3.2 Scalability

The scalability of algorithms in the non-adaptive case and the adaptive case are reported in Fig. 4 and Fig. 5 respectively. GBT is the building time of reversely reachable sets.

- **Non-adaptive Case**

According to Fig. 4, the running time of IM and CD is almost the same as the GBT, since computing the allocation in 100 users will not take too much time. With the increase of the network size, the gap between 2CD and GBT decreases. In the smallest dataset Wiki-Vote, the running time of 2CD is about 20 times that of GBT. However, the ratio becomes less than 1.3 in the largest dataset soc-Livejournal. The reason is that the computation cost of the hyper-graph is high, while the execution of the 2CD algorithm is relatively efficient. It is worth noting that, the RF algorithm has the least running time and the best scalability, since RF does not need to build the hyper-graph. Thus, the simple two-stage algorithm RF outperforms IM and CD in both influence spread and scalability.

- **Adaptive Case**

It is shown in Fig. 5 that the running time of the adaptive algorithms follows the sequence: Ada+CD > Ada+MGS > Ada+GS > A-Greedy > Ada. We next explain the sequence in an increasing order. The running time of Ada is the smallest since it does not need to consider the allocation

in neighbors. A-Greedy computes the allocation in both stages, so the running time is higher than Ada. The running time of Ada+GS is larger than that of A-Greedy, since fine-grained discounts incur more sophisticated computation. It is not a surprise to see that Ada+MGS takes more time than Ada+GS, since enumeration is applied in Ada+MGS. As for Ada+CD, to estimate the benefit of each action, numerous coordinate descent algorithms are carried out in stage 2, incurring tremendous iterations. Thus, Ada+CD is less efficient than Ada+MGS. Similar to the non-adaptive case, the gap between algorithms and GBT is decreasing as the network size grows, due to the same reason in the non-adaptive case. The running time of LP is almost in the same scale over the four satasets, since it is not based on the hyper-graph and only needs to solve an LP problem.

From the results in both cases, we find that most time is spent on building the hyper-graph. The execution time (GBT not included) of the proposed algorithms is less than one hour in all the cases except one point ($B = 50$ in Fig. 4(d)). Generally, the running time of 2CD is larger than Ada+MGS while smaller than Ada+CD. Thus, we can roughly obtain the sequence of running time of the proposed algorithms: Ada+CD > 2CD > Ada+MGS > Ada+GS.

7.3.3 Sensitivity

We also test the sensitivity of our proposed algorithms with respect to different settings of seed probability functions. To this end, we introduce a second setting with different portion of seed probability functions. The previous setting

TABLE 3
Sensitivity to the seed probability function

Dataset	B	2CD			Ada+CD			Ada+GS			Ada+MGS		
		S1	S2	Reduction	S1	S2	Reduction	S1	S2	Reduction	S1	S2	Reduction
Wiki-Vote	10	67	57	14.9%	110	65	40.9%	79	70	11.4%	159	135	15.1%
	20	146	98	32.9%	182	121	33.5%	125	97	22.4%	241	218	9.5%
	30	171	119	30.4%	228	162	28.9%	201	170	15.4%	309	274	11.3%
	40	239	177	25.9%	290	184	36.6%	251	206	17.9%	349	330	5.4%
	50	293	208	29%	334	205	38.6%	292	272	6.8%	397	379	4.5%
Condmat	10	411	252	38.7%	618	321	48.1%	579	407	29.7%	665	566	14.9%
	20	616	390	35.6%	953	558	41.4%	820	593	27.7%	985	872	11.5%
	30	803	595	25.9%	1122	733	34.7%	1013	747	26.3%	1212	1099	9.3%
	40	934	686	26.6%	1333	814	38.9%	1184	881	25.6%	1483	1282	13.6%
	50	1132	863	23.8%	1465	859	41.3%	1376	1029	25.2%	1685	1498	11.1%
Dblp	10	358	251	29.9%	685	266	46.6%	543	345	36.5%	859	703	18.2%
	20	802	581	27.6%	1287	553	57%	891	621	30.3%	1501	1176	21.7%
	30	1019	724	28.9%	1852	851	54%	1332	880	33.9%	1953	1612	17.5%
	40	1308	954	27.1%	2261	983	56.5%	1735	1322	23.8%	2403	2119	11.8%
	50	1644	1201	26.9%	2608	1191	54.3%	2125	1607	24.4%	2793	2440	12.6%
Livejournal	10	1564	1073	31.4%	4591	2524	45%	4481	2626	41.4%	4610	3632	21.2%
	20	3193	2333	26.9%	6405	4325	32.5%	6040	4159	31.1%	7199	6843	4.9%
	30	5340	3222	39.7%	8686	5148	40.7%	7067	5114	27.6%	9523	8785	7.7%
	40	6935	5462	21.2%	10831	6590	39.2%	7969	6776	15%	11310	10793	4.6%
	50	7464	6280	15.9%	12913	7246	56.1%	9000	7541	16.2%	12781	12719	0.5%

TABLE 4
The impact of the friendship paradox phenomenon

Dataset	B	Non-adaptive						Adaptive					
		$|X| = 100$			$|X| = 1000$			$|X| = 100$			$|X| = 1000$		
		CD	2CD	Increase	CD	2CD	Increase	Ada	Ada+MGS	Increase	Ada	Ada+MGS	Increase
Wiki-Vote	10	27	67	40	34	50	16	68	159	91	149	205	56
	20	47	146	99	60	106	46	98	241	143	260	301	41
	30	66	190	124	84	149	65	121	309	188	336	382	46
	40	85	239	154	110	220	110	139	349	210	398	439	41
	50	101	293	192	128	270	142	146	397	251	451	489	38
Condmat	10	67	411	344	83	135	52	206	665	459	453	652	199
	20	130	616	486	156	295	139	292	985	693	725	1039	314
	30	182	803	621	234	558	324	348	1272	924	935	1314	379
	40	231	934	703	306	744	438	375	1483	1108	1105	1570	465
	50	276	1132	856	377	762	385	375	1685	1310	1258	1774	516
Dblp	10	69	358	289	75	251	176	188	859	671	412	1126	714
	20	131	802	671	146	680	534	275	1501	1226	657	2007	1350
	30	185	1019	798	213	996	783	332	1953	1621	850	2763	1913
	40	228	1308	1080	281	1157	876	360	2403	2043	1011	3417	2406
	50	268	1644	1376	346	1424	1328	360	2793	2433	1159	4057	2898
Livejournal	10	138	1564	1426	237	1508	1271	408	4436	4028	980	24863	23883
	20	256	3193	2937	425	2523	2098	550	7199	6649	1603	30029	28426
	30	363	5340	4977	580	5389	4809	680	9523	8843	2089	34059	31970
	40	464	6935	6471	713	6101	5388	744	11310	10566	2465	38132	35667
	50	556	7464	6908	863	7261	6398	749	12781	12032	2788	41280	38492

in Section 7.1 is denoted as Setting 1 (S1). In Setting 2 (S2), 65% users are assigned with $p_u(c_u) = 2c_u − c_u^2$, and 20% users with $p_u(c_u) = c_u$, and 15% users with $p_u(c_u) = c_u^2$. The algorithms are run again in Setting 2.

Table 3 reports the influence spread of the proposed algorithms with $\alpha = 1.0$. As can be seen, the influence spread of the four algorithms decreases in setting 2. This phenomenon indicates that users are harder to satisfy under Setting 2. In terms of the ability to cope with the change of seed probability functions, 2CD shows even better performance than Ada+CD and comparable performance with Ada+GS. The possible reason is that 2CD makes refinement over a large amount of users in $N(S)$, while Ada+CD and Ada+GS only allocate discounts to neighbors of one agent each time. Thus, it is easier for 2CD to find an alternative user to seed when the seed probability function of a user becomes not favorable. However, Ada+MGS shows the best performance in coping with the change of settings. The

explanation is that the enumeration process helps find good action combinations in Setting 2.

7.3.4 Impact of the Friendship Paradox Phenomenon

In this part, we evaluate the impact of the FP phenomenon. The algorithms are evaluated under two settings, i.e., $|X| = 100$ and 1000. To show the effect of the FP phenomenon, we compare the influence spread between one-stage algorithms and two-stage algorithms. In the non-adaptive case, CD and 2CD are selected for experiment. In the adaptive case, the only one-stage algorithm Ada is tested and Ada+MGS is selected due to its impressive performance.

The influence spread is shown in Table 4. Regardless of the size of X, two-stage algorithms show larger influence spread than one-stage algorithms by exploiting the influential neighborhood. We next focus on the influence spread of algorithms in the larger $|X|$. For the two one-stage algorithms, when $|X| = 1000$, the influence spreads

both become larger, since the number of influential users is likely to be larger in a larger X. However, the performance of two-stage algorithms is different. When $|X| = 1000$, the influence spread of 2CD becomes smaller, while Ada+MGS shows even better performance. We further compare the increase of influence spread brought by two-stage algorithms. In the two relatively smaller datasets, i.e., Wiki-Vote and Ca-Condmat, the increases of 2CD and Ada+MGS both become smaller when $|X| = 1000$. However, in Dblp and LiveJournal, the increase of Ada+MGS is larger when $|X| = 1000$, while the increase of 2CD is still smaller. This observation indicates that Ada+MGS has a better ability to utilize the friendship paradox phenomenon.

8 CONCLUSION AND FUTURE WORK

This paper studies the influence maximization problem with limited initially accessible users. To overcome the access limitation, we propose a new two-stage seeding model with the FP phenomenon embedded, where neighbors are further seeded. Based on this model, we solve the limited influence maximization problem under both non-adaptive and adaptive cases. In the non-adaptive case, we examine the properties of this problem and establish a two-stage coordinate descent framework to determine the discount allocation in two stages. In the adaptive case, we first consider the discrete-continuous setting and design the adaptive greedy algorithm with theoretical guarantee. Then, in the discrete-discrete setting, the allocation in stage 2 is considered to be discrete. Accordingly, two algorithms are proposed based on greedy selection. Finally, extensive experiments are carried out on real-world datasets to evaluate the performance of the proposed algorithms. Moreover, our work is only a primary study into the two-stage IM, since the derivation of diffusion probabilities is not considered. While by serving as a subroutine, our study would continue to benefit the design of effective online algorithms which consider the learning of diffusion parameters.

In the future, we would like to devote to finding better allocations than the convergent one to improve the coordinate descent algorithm. In the adaptive case, it is implicitly assumed that we have enough time to observe the whole diffusion process, which may take lots of time and thus impractical. It is worthwhile to study the problem when only part of the diffusion is observed in each round. In footnote 3, although we proposed solutions to the case of unified budget, it is still necessary to treat it as an individual problem and more effective algorithms could be designed.

REFERENCES

[1] D. Kempe, J. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," in *Proc. Int. Conf. Knowl. Discov. Data Mining, KDD*. ACM, 2003, pp. 137–146.

[2] S. Bharathi, D. Kempe, and M. Salek, "Competitive influence maximization in social networks," *Internet Netw. Econ.*, pp. 306–311, 2007.

[3] W. Chen, Y. Yuan, and L. Zhang, "Scalable influence maximization in social networks under the linear threshold model," in *Proc. Int. Conf. Data Mining, ICDM*. IEEE, 2010, pp. 88–97.

[4] P. Domingos and M. Richardson, "Mining the network value of customers," in *Proc. Int. Conf. Knowl. Discov. Data Mining, KDD.* ACM, 2001, pp. 57–66.

[5] A. Goyal, F. Bonchi, and L. V. Lakshmanan, "A data-based approach to social influence maximization," *Proc. VLDB Endow.*, vol. 5, no. 1, pp. 73–84, 2011.

[6] H. Nguyen and R. Zheng, "On budgeted influence maximization in social networks," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 6, pp. 1084–1094, 2013.

[7] Y. Tang, Y. Shi, and X. Xiao, "Influence maximization in near-linear time: A martingale approach," in *Proc. Int. Conf. Manage. Data, SIGMOD*. ACM, 2015, pp. 1539–1554.

[8] Y. Yang, X. Mao, J. Pei, and X. He, "Continuous influence maximization: What discounts should we offer to social network users?" in *Proc. Int. Conf. Manage. Data, SIGMOD*. ACM, 2016, pp. 727–741.

[9] J. Yuan and S.-J. Tang, "Adaptive discount allocation in social networks," in *Proc. Int. Symp. Mobile Ad Hoc Netw. Comput., MOBIHOC*. ACM, 2017, pp. 1–10.

[10] A. Badanidiyuru, C. Papadimitriou, A. Rubinstein, L. Seeman, and Y. Singer, "Locally adaptive optimization: Adaptive seeding for monotone submodular functions," in *Proc. Annu. ACM-SIAM Symp. Discrete Algorithms*. SIAM, 2016, pp. 414–429.

[11] T. Horel and Y. Singer, "Scalable methods for adaptively seeding a social network," in *Proc. Int. Conf. World Wide Web*. ACM, 2015, pp. 441–451.

[12] A. Rubinstein, L. Seeman, and Y. Singer, "Approximability of adaptive seeding under knapsack constraints," in *Proc. ACM Conf. Econ. Comput.* ACM, 2015, pp. 797–814.

[13] L. Seeman and Y. Singer, "Adaptive seeding in social networks," in *Proc. Annu. Symp. Found. Comput. Sci., FOCS*. IEEE, 2013, pp. 459–468.

[14] S. L. Feld, "Why your friends have more friends than you do," *Amer. J. Sociol.*, vol. 96, no. 6, pp. 1464–1477, 1991.

[15] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions-i," *Math. Program.*, vol. 14, no. 1, pp. 265–294, 1978.

[16] U. Feige, "A threshold of ln n for approximating set cover," *J. of the ACM*, vol. 45, no. 4, pp. 634–652, 1998.

[17] W. Chen, A. Collins, R. Cummings, T. Ke, Z. Liu, D. Rincon, X. Sun, Y. Wang, W. Wei, and Y. Yuan, "Influence maximization in social networks when negative opinions may emerge and propagate," in *Proc. Int. Conf. Data Mining, ICDM*. IEEE, 2011, pp. 379–390.

[18] K. Jung, W. Heo, and W. Chen, "Irie: Scalable and robust influence maximization in social networks," in *Proc. Int. Conf. Data Mining, ICDM*. IEEE, 2012, pp. 918–923.

[19] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance, "Cost-effective outbreak detection in networks," in *Proc. Int. Conf. Knowl. Discov. Data Mining, KDD*. ACM, 2007, pp. 420–429.

[20] G. Li, S. Chen, J. Feng, K.-l. Tan, and W.-s. Li, "Efficient location-aware influence maximization," in *Proc. Int. Conf. Manage. Data, SIGMOD*. ACM, 2014, pp. 87–98.

[21] Y. Tang, X. Xiao, and Y. Shi, "Influence maximization: Near-optimal time complexity meets practical efficiency," in *Proc. Int. Conf. Manage. Data, SIGMOD*. ACM, 2014, pp. 75–86.

[22] W. Chen, Y. Wang, and S. Yang, "Efficient influence maximization in social networks," in *Proc. Int. Conf. Knowl. Discov. Data Mining, KDD.* ACM, 2009, pp. 199–208.

[23] W. Chen, L. V. Lakshmanan, and C. Castillo, "Information and influence propagation in social networks," *Synthesis Lectures Data Manage.*, vol. 5, no. 4, pp. 1–177, 2013.

[24] B. Fotouhi, N. Momeni, and M. G. Rabbat, "Generalized friendship paradox: An analytical approach," in *Int. Conf. on Social Inform.* Springer, 2014, pp. 339–352.

[25] J. B. Kramer, J. Cutler, and A. Radcliffe, "The multistep friendship paradox," *Amer. Math. Monthly*, vol. 123, no. 9, pp. 900–908, 2016.

[26] S. Lattanzi and Y. Singer, "The power of random neighbors in social networks," in *Proc. Int. Conf. Web Search and Data Mining*. ACM, 2015, pp. 77–86.

[27] D. Golovin and A. Krause, "Adaptive submodularity: Theory and applications in active learning and stochastic optimization," *J. of Artif. Intelli. Res.*, vol. 42, pp. 427–486, 2011.

[28] S. Tang, "No time to observe: Adaptive influence maximization with partial feedback," in *Proc. Int. Joint Conf. Artif. Intelli.* AAAI, 2017, pp. 3908–3914.

[29] A. Krause and C. Guestrin, *A note on the budgeted maximization of submodular functions.* Carnegie Mellon University. Center for Automated Learning and Discovery, 2005.

[30] J. Leskovec and A. Krevl, "SNAP Datasets: Stanford large network

dataset collection," http://snap.stanford.edu/data, Jun. 2014.

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

[32] G. Tong, W. Wu, S. Tang, and D.-Z. Du, "Adaptive influence maximization in dynamic social networks," *IEEE/ACM Trans. Netw.*, vol. 25, no. 1, pp. 112–125, 2017.

Xinbing Wang received the B.S. degree (with hons.) in automation from Shanghai Jiao Tong University, China, in 1998, the M.S. degree in computer science and technology from Tsinghua University, China, in 2001, and the Ph.D. degree with a major in electrical and computer engineering and minor in mathematics from North Carolina State University, in 2006. Currently, he is a Professor in the Department of Electronic Engineering, and Department of Computer Science, Shanghai Jiao Tong University, China. Dr. Wang has been an Associate Editor for IEEE/ACM TRANSACTIONS ON NETWORKING, IEEE TRANSACTIONS ON MOBILE COMPUTING, and ACM Transactions on Sensor Networks. He has also been the Technical Program Committees of several conferences including ACM MobiCom 2012,2014, ACM MobiHoc 2012-2017, IEEE INFOCOM 2009-2017.

Chen Feng received his B.E. degree in Communication Engineering from Tianjin University, China, in 2016. He is currently pursuing the Ph.D. degree in Electronic Engineering at Shanghai Jiao Tong University, Shanghai, China. His current research interests are in the area of social networks and information diffusion.
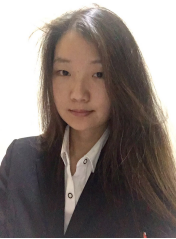
Luoyi Fu received her B.E. degree in Electronic Engineering from Shanghai Jiao Tong University, China, in 2009 and Ph.D. degree in Computer Science and Engineering in the same university in 2015. She is currently an Associate Professor in Department of Computer Science and Engineering in Shanghai Jiao Tong University. Her research of interests are in the area of social networking and big data, scaling laws analysis in wireless networks, connectivity analysis and random graphs.

Feilong Tang received the Ph.D. degree in computer science and technology from Shanghai Jiao Tong University, in 2005. He was a Japan Society for the Promotion of Science (JSPS) Postdoctoral Research Fellow. Currently, he works with the School of Software, Shanghai Jiao Tong University. His research interests include cognitive and sensor networks, protocol design for communication networks, and pervasive and cloud computing. He has published more than 100 papers in journals and international conferences and works as a PI of many projects such as National Natural Science Foundation of China (NSFC) and National High-Tech R&D Program (863 Program) of China.

Bo Jiang received the B.S. degree in Electronic Engineering from Tsinghua University, Beijing, China in 2006, the M.S. degree in Electrical and Computer Engineering from the University of Massachusetts Amherst in 2008 and the Ph.D. degree in Computer Science from the same university. He is currently an Associate Professor in the John Hopcroft Center in Shanghai Jiao Tong University. His research includes network modeling and analysis, wireless networks, and network science.

Guihai Chen received the B.S. degree from Nanjing University, the M.E. degree from Southeast University, and the Ph.D. degree from The University of Hong Kong. He visited the Kyushu Institute of Technology, in 1998, as a Research Fellow, and the University of Queensland, in 2000, as a Visiting Professor. From 2001 to 2003, he was a Visiting Professor with Wayne State University. He is currently a Distinguished Professor and a Deputy Chair with the Department of Computer Science, Shanghai Jiao Tong University. His research interests include wireless sensor networks, peer-to-peer computing, and performance evaluation. He has served on technical program committees of numerous international conferences.

Haisong Zhang received his B.E. degree in Computer Science and Technology from Xidian University, China, in 2011 and the M.E. Degree in Software Engineering from the same university in 2014. He is currently a Senior Researcher in Tencent AI Lab. His research interests include natural language understanding and dialogue system etc..