Supplementary Material for: Neighborhood Matters: Influence Maximization in Social Networks with Limited Access

Due to space limitation, some illustrations, proofs and experimental results are omitted in the main paper, and we provide them in this supplemental file for completeness.

1 Seeding Examples

In this section, for readers' comprehension, we present two seeding examples to illustrate the non-adaptive seeding process and the adaptive process respectively. For ease of illustration, we assume $B_1 = B_2 = 1$ and the discount rate in the adaptive case is $D = \{0.5, 1.0\}$.

1.1 Non-adaptive Seeding Example

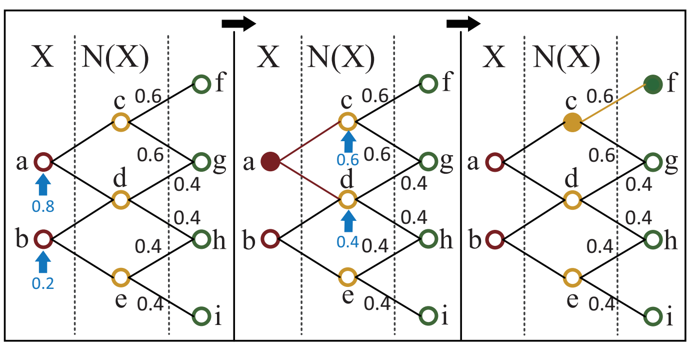


Fig. 1. An example of the non-adaptive seeding process.

In the recruitment stage, user a seems to be more profitable, since a is able to reach the influential user c. Thus, we allocate discount 0.8 to user a and 0.2 to user b in stage 1. Budget B_1 is used up. Suppose user a becomes an agent while b does not. Then, we reach users c and d via user a. In the trigger stage, we distribute budget $B_2 = 1$ to newly reachable users by allocating discount 0.6 to user c and 0.4 to user d. User c becomes the seed and then influence diffusion starts from it. Finally, user f gets influenced.

1.2 Adaptive Seeding Example

In the adaptive case, initially accessible users are seeded sequentially. In the first round, we adopt the action $(a, 0.5)$, i.e., seeding user a with discount 0.5. Unfortunately, a refuses the discount. Then, we move to the second round, where the remaining budget B_1 is still 1. We adopt the action $(b, 1.0)$ and user b accepts the discount. Then, b's neighbors d and e become reachable. Both d and e are

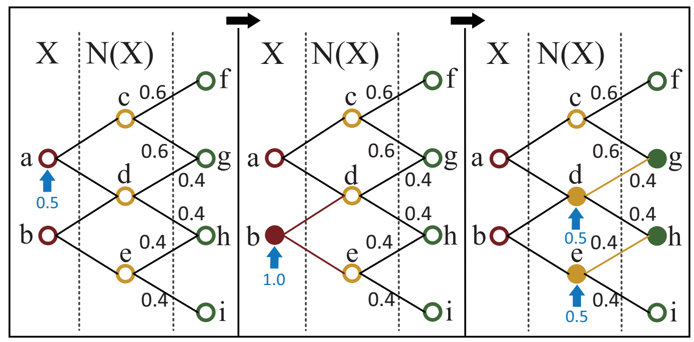


Fig. 2. An example of the adaptive seeding process.

provided with discount 0.5 from budget B_2. Suppose d and e both become the seed and the influence diffusion starts from them. Finally, users g and h get influenced.

2 Missing Proofs

In this section, we present the omitted proofs for Theorems and Lemmas.

2.1 Proof of Lemma 1

We prove lemma 1 by the probabilistic method. Let $d(\cdot)$ denote the degree of a node and $\bar{d}(\cdot)$ denote the average degree of nodes in a set. The expected average degree of nodes in X is

$$E[\bar{d}(X)] = E\left[\frac{\sum_{v \in X} d(v)}{|X|}\right] = \frac{1}{|X|}\sum_{v \in X} E[d(v)] = E[d(v)].$$

Recall that nodes in V are randomly selected into X with probability $p \to 0$. For node v in V, the probability that it is in $N(X)$ is $(1-p)[1-(1-p)^{d(v)}]$, i.e., v is not in X, but at least one of its neighbors is in X. Since $p \to 0$, the probability can be approximated by $p(1-p)d(v)$. Thus, the size of the neighborhood of X can be denoted as $|N(X)| = \sum_{v \in V} p(1-p)d(v)$. The sum of degrees of nodes in $N(X)$ is $\sum_{v \in V} p(1-p)d^2(v)$. The expected average degree of nodes in $N(X)$ is

$$E[\bar{d}(N(X))] = E\left[\frac{\sum_{v \in V} p(1-p)d^2(v)}{\sum_{v \in V} p(1-p)d(v)}\right] = E\left[\frac{\sum_{v \in V} d^2(v)}{\sum_{v \in V} d(v)}\right]$$

Applying Cauchy-Schwartz inequality, we have $\sum_{v \in V} d^2(v) \geq \frac{1}{n}[\sum_{v \in V} d(v)]^2$. Thus, $E[\bar{d}(N(X))] \geq$

$E\left[\frac{1}{n}\sum_{v\in V} d(v)\right] = E[d(v)]$. Therefore, $E[\bar{d}(N(X))] \geq E[\bar{d}(X)]$. This completes the proof.

2.2 Proof of Lemma 2

We first show the NP-hardness of the discrete two-stage non-adaptive influence maximization by reduction from the NP-complete *Set Cover* problem. Consider an arbitrary instance of Set Cover. Let $A = \{A_1, A_2, \cdots, A_m\}$ be a family of subsets of a ground set $U = \{u_1, u_2, \cdots, u_n\}$, satisfying $\bigcup_{i=1}^{m} A_i = U$. The problem is whether there exists a subfamily $C \subseteq A$, $|C| = k$, whose union is U.

(1) We proceed to construct a discrete non-adaptive influence maximization problem corresponding to the Set Cover problem. Define a graph G where there is only one node x in X. For each subset A_i, there is a node i corresponding to it, with a directed edge from x to i. For each element u_j, there is a node j corresponding to it. If $u_j \in A_i$, then there is an edge between i and j. All edges are associated with probability 1. The budget is $k+1$. In the two-stage setting, one budget must be spent on initially reachable user x to reach its neighbors. Thus, the Set Cover problem is equivalent to selecting k nodes in $N(x)$ to influence the n nodes corresponding to U.

(2) If there is a solution to the discrete maximization problem, then we can solve the Set Cover problem by selecting corresponding k subsets in A to cover the n elements in U.

(3) It is obvious that the reduction from Set Cover can be completed in polynomial time by traversing A and U.

Thus, we prove that the discrete case is NP-hard. We can show that our continuous non-adaptive influence maximization is NP-hard by reduction from the discrete case, which is trivial and thus omitted. Intuitively, the discrete IM is only a special case of our continuous setting.

2.3 Proof of Lemma 3

If $T_1 = T_2$, Lemma 3 holds definitely. We focus on $T_1 \subsetneqq T_2$.

To begin with, we prove a simple case where $T_2 = T_1 \cup \{v\}$. Assume the optimal discount allocation in T_1 is C_2^*. Since all users in T_1 are also in T_2 and the budget is the same, C_2^* is a feasible allocation in T_2 where the discount of user v is 0 and the discounts of other users are the same as C_2^*. Then, we have $Q(C_2^*; T_2) = \max Q(C_2; T_1)$. Due to the optimality of $\max Q(C_2; T_2)$, we have $\max Q(C_2; T_2) \geq Q(C_2^*; T_2) = \max Q(C_2; T_1)$, i.e. $\max Q(C_2; T_1 \cup \{v\}) \geq \max Q(C_2; T_1)$. By the transitivity of $\geq$, we know that $\max Q(C_2; T_2) \geq \max Q(C_2; T_1)$ holds for all $T_1 \subsetneqq T_2$. Thus, the proof of Lemma 3 is completed.

2.4 Proof of Theorem 1

(1) We start from a simple case, where only one element is different between C_2 and C_2'. Then, the result can be extended to general cases by the transitivity of $\geq$.

Assume that only the discount of user $u \in N(S)$ is different. u gets discount c_u in C_2 while c_u' in C_2' and $c_u \geq c_u'$. By the definition of $Q(C_2; N(S))$, it can be written as

$$Q(C_2; N(S)) = \sum_{T \subseteq N(S)\backslash\{u\}} P_r(T; C_2, N(S)\backslash\{u\}) \cdot \Big\{[1 - p_u(c_u)]I(T) + p_u(c_u)I(T \cup \{u\})\Big\}.$$

After a simple calculation, we have

$$Q(C_2; N(S)) - Q(C_2'; N(S)) = \sum_{T \subseteq N(S)\backslash\{u\}} P_r(T; C_2, N(S)\backslash\{u\}) \cdot \Big\{[p_u(c_u) - p_u(c_u')][I(T \cup \{u\}) - I(T)\Big\}.$$

Since $I(\cdot)$ is nondecreasing under the independent cascade model and $p_u(\cdot)$ is nondecreasing as well by assumption, we have $Q(C_2; N(S)) \geq Q(C_2'; N(S))$.

(2) Following similar technique, we prove the result when there is only one element different between C_1 and C_1'. Assume the different element is u and u gets discount c_u in C_1 while c_u' in C_1' with $c_u \geq c_u'$. By the definition of $f(C_1; X)$, $f(C_1; X)$ can be written as

$$f(C_1; X) = \sum_{S \subseteq X\backslash\{u\}} P_r(S; C_1, X \setminus \{u\}) \Big\{p_u(c_u) \max Q(C_2; N(S \cup \{u\})) + [1 - p_u(c_u)] \max Q(C_2; N(S))\Big\}.$$

After a simple calculation, we have

$$f(C_1; X) - f(C_1'; X) = \sum_{S \subseteq X\backslash\{u\}} P_r(S; C_1, X \setminus \{u\}) \Big\{[p_u(c_u) - p_u(c_u')][\max Q(C_2; N(S \cup \{u\})) - \max Q(C_2; N(S))]\Big\}.$$

Note that the budget in stage 2 is the same under C_1 and C_1'. According to Lemma 3, $\max Q(C_2; N(S \cup \{u\})) - \max Q(C_2; N(S)) \geq 0$. According to the monotonicity of $p_u(\cdot)$ and $c_u \geq c_u'$, we have $f(C_1; X) \geq f(C_1'; X)$.

Combining case (1) and case (2), we complete the proof.

2.5 Proof of Lemma 4

It is easy to verify the adaptive monotonicity since under any realization, since the influence spread will not decrease when more users are seeded.

To prove the adaptive submodularity, according to its definition, it is equivalent to prove that $\Delta(y|\psi) \geq \Delta(y|\psi')$ holds for any $\psi \subseteq \psi'$ and $y \in Y \setminus dom(\psi)$. Note that ψ is a process with sequence recording the actions adopted. $\psi \subseteq \psi'$ means ψ is a subprocess of ψ', i.e., ψ is a history of ψ' and ψ' went through all what ψ has experienced. Therefore, the seeding result of nodes and states of edges observed by ψ are the same in ψ'.

We first introduce some notations. The diffusion realization ϕ (resp. ϕ') is a function of the states of edges, which are denoted as a series of random variables $X = \{X_{ij}, (i,j) \in E\}$ (resp. $X' = \{X'_{ij}, (i,j) \in E\}$).

We attempt to define a coupled distribution $\rho((\lambda, \phi), (\lambda', \phi))$ over two pairs of realizations $(\lambda, \phi) \sim \psi$ and $(\lambda', \phi') \sim \psi'$. Recall the definition of $(\lambda, \phi) \sim \psi$ that λ is consistent with the partial seeding realization observed by ψ and ϕ is consistent with the states of edges explored under ψ. Since $(\lambda, \phi) \sim \psi$, $(\lambda', \phi') \sim \psi'$, and $\psi \subseteq \psi'$, the states of nodes and edges observed by ψ are the same in (λ, ϕ) and (λ', ϕ'). Then, the diffusion brought by action y (i.e., $\Delta(y|\psi)$) is only dependent on the states of unknown edges. Thus, we will reduce the domain of ρ to ϕ and ϕ'. We define the coupled distribution ρ in terms of a joint distribution $\hat{\rho}$ on $X \times X'$, where $\phi = \phi(X)$ and $\phi' = \phi'(X')$ are the diffusion realizations induced by the two distinct sets of random edge states respectively. Recall that the domain of ρ is reduced to ϕ and ϕ'. Hence, $\rho((\lambda, \phi(X)), (\lambda', \phi(X'))) = \hat{\rho}(X, X')$.

We say the seeding process ψ observes an edge if it is explored and the state is revealed. For any edge (i, j) observed by ψ (resp. ψ'), its state X_{ij} (resp. X'_{ij}) is deterministic. Recall that the states of edges observed by ψ are the same in ϕ and ϕ', since $\psi \subseteq \psi'$. We will construct $\hat{\rho}$ so that the states of all edges unobserved by both ψ and ψ' are the same in X and X', i.e., $X_{ij} = X'_{ij}$, otherwise $\hat{\rho}(X, X') = 0$. The above constraints allow us to select X_{ij} whose edges are unobserved by ψ. We select such variables independently. Hence for all (X, X') satisfying the above constraints, we have

$$\hat{\rho}(X, X') = \prod_{(i,j) \text{ unobserved by } \psi} p_{ij}^{X_{ij}} (1 - p_{ij})^{1 - X_{ij}},$$

otherwise $\hat{\rho}(X, X') = 0$.

We next try to prove that the following formula holds for any $((\lambda, \phi), (\lambda', \phi')) \in support(\rho)$,

$$\begin{aligned} &\hat{\sigma}(dom(\psi') \cup \{y\}, (\lambda', \phi')) - \hat{\sigma}(dom(\psi'), (\lambda', \phi')) \leq \\ &\hat{\sigma}(dom(\psi) \cup \{y\}, (\lambda, \phi)) - \hat{\sigma}(dom(\psi), (\lambda, \phi)). \end{aligned} \quad (1)$$

Let set B denote $\sigma(dom(\psi), (\lambda, \phi))$, D denote $\sigma(dom(\psi) \cup \{y\}, (\lambda, \phi))$, B' denote $\sigma(dom(\psi'), (\lambda', \phi'))$ and D' denote $\sigma(dom(\psi') \cup \{y\}, (\lambda', \phi'))$. We will first show that $B \subseteq B'$. For any node $i \in B$, there exists a path from some node $j \in dom(\psi)$ to it. Each edge in this path is observed to be live. Since $\psi \subseteq \psi'$ and $(\lambda, \phi) \sim \psi$, $(\lambda', \phi') \sim \psi'$, the edge observed to be live in ψ must be live as well in ψ', and j must also be a seed in ψ'. Then, there is also a path from i to j under (λ', ϕ'). Thus, $B \subseteq B'$.

We proceed to prove formula (1). Since $\psi \subseteq \psi'$, we have $dom(\psi) \subseteq dom(\psi')$ and thus $N(\bigcup_{p \in dom(\psi)} v(p)) \subseteq N(\bigcup_{p \in dom(\psi')} v(p))$. Therefore, nodes newly reached by $v(y)$ under ψ are part of those under ψ', that is, $N(v(y)) \setminus N(\bigcup_{p \in dom(\psi')} v(p)) \subseteq N(v(y)) \setminus N(\bigcup_{p \in dom(\psi)} v(p))$. Note that the budget allocated to newly reached nodes under ψ and ψ' is the same, because the budget drawn from B_2 only depends on the intrinsic property of $v(y)$ itself. Furthermore, $((\lambda, \phi), (\lambda', \phi')) \in support(\rho)$, the states of unobserved edges are the same. Thus, according to Lemma 3 and $B \subseteq B'$, we can see that $D \setminus B$ is a superset of $D' \setminus B'$. In addition, $\hat{\sigma} = |\sigma|$, $B \subseteq D$ and $B' \subseteq D'$, hence formula (1) holds.

For $((\lambda, \phi), (\lambda', \phi')) \notin support(\rho)$, $\rho((\lambda, \phi), (\lambda', \phi')) = 0$. Then, summing over $((\lambda, \phi), (\lambda', \phi'))$ in $support(\rho)$ and not in $support(\rho)$, we have

$$\begin{aligned} &\sum_{((\lambda,\phi),(\lambda',\phi'))} \rho((\lambda, \phi), (\lambda', \phi'))(\hat{\sigma}(dom(\psi') \cup \{y\}, (\lambda', \phi')) \\ &\qquad - \hat{\sigma}(dom(\psi'), (\lambda', \phi'))) \leq \\ &\sum_{((\lambda,\phi),(\lambda',\phi'))} \rho((\lambda, \phi), (\lambda', \phi'))(\hat{\sigma}(dom(\psi) \cup \{y\}, (\lambda, \phi)) \\ &\qquad - \hat{\sigma}(dom(\psi), (\lambda, \phi))). \end{aligned} \quad (2)$$

Note that $p((\lambda, \phi)|\psi) = \sum_{(\lambda', \phi')} \rho((\lambda, \phi), (\lambda', \phi'))$ and $p((\lambda', \phi')|\psi') = \sum_{(\lambda, \phi)} \rho((\lambda, \phi), (\lambda', \phi'))$. We sum over (λ, ϕ) in the left side of formula (2) and (λ', ϕ') in the right side. Combining the definition of $\Delta(y|\psi)$ and $\Delta(y|\psi')$, we have $\Delta(y|\psi) \geq \Delta(y|\psi')$.

2.6 Proof of Lemma 5

We would like to prove this lemma by induction. Let S_m denote the first m seeds selected by π^{greedy} and R_m denote the first m seeds selected by $\pi^{\text{greedy}}_{\text{relaxed}}$.

(i) Let us consider the basic case, $m = 1$, i.e. the first seed. It is easy to see that $\pi^{\text{greedy}}_{\text{relaxed}}$ will choose user u that maximizes $\frac{E[\hat{\sigma}(y,(\Lambda,\Phi))]}{d_{\min}(u)}$, where $y = (u, d_{\min}(u))$.

As for π^{greedy}, we probe $y^* = \text{argmax}_{y \in Y} \frac{\Delta(y|\psi_p)}{d(y)} = \text{argmax}_{y \in Y} \frac{E[\hat{\sigma}(y,(\Lambda,\Phi))]}{d(y)}$. Each time, the selected action is refused or accepted. We accordingly delete the action and move to the next round or get a seed.

The action space Y can be divided into a union of disjoint action subsets $Y := \bigcup_{v \in X} Y_v$, where Y_v is the set of actions about user v, i.e., $Y_v = \{y | v(y) = v\}$. We next show that for each action subset Y_v, there is no need to consider actions whose discounts are not $d_{\min}(v)$. Due to the greedy policy π^{greedy}, user v will be probed with Y_v from the smallest discount. For actions with discount less than $d_{\min}(v)$, v will reject it. When the discount becomes $d_{\min}(v)$, v becomes a seed and remaining polices are abandoned. Thus, it is equivalent to select actions from $Y^* = \{(v, d_{\min}(v)) | v \in X\}$. Then, the selection becomes selecting an action from Y^* that maximizes $\frac{E[\hat{\sigma}(y,(\Lambda,\Phi))]}{d_{\min}(v(y))}$, which is the same as $\pi^{\text{greedy}}_{\text{relaxed}}$. Thus, the two algorithms will yield the same first seed.

(ii) Assume that S_m and R_m are the same when $m = k$, we proceed to the case $m = k + 1$. Given partial seeding process ψ_p, the seed selected by $\pi^{\text{greedy}}_{\text{relaxed}}$ is the user u that maximizes $\frac{\Delta((u,d_{\min}(u))|\psi_p)}{d_{\min}(u)}$. In terms of π^{greedy}, for each user u, $\Delta(y|\psi_p)$ is the same for any action y about u, i.e. $v(y) = u$, since u is assumed to be the seed when calculating $\Delta(y|\psi_p)$ and the budget allocated in stage 2 only depends on user u itself. Following similar arguments in (i), we derive the $(k+1)$-th seed selected by maximizing $\frac{E[\hat{\sigma}(y,(\Lambda,\Phi))]}{d_{\min}(v(y))}$. Thus, the $(k+1)$-th seed is the same in two algorithms.

Combining (i) and (ii), we complete the proof of Lemma 5.

2.7 Proof of Theorem 2

We next analyze the budget used in stage 1 in the relaxed setting. Let us consider the selection of the last action, if the remaining budget is smaller than the minimum desired discounts d_v of all the remaining users, then no one will accept the discount and the remaining budget can not be used. In the extreme case, all the remaining users desire discount 1. Thus, the budget used in stage 1 is at least $B_1 - 1$. As noted in Section 5, with proper initial allocation and enough iterations, the local optimum of the coordinate descent algorithm in stage 2 could be eliminated. Let $\pi^{\text{greedy}}_{\text{relaxed}}$ denote the optimal action of the relaxed setting. According to Theorem A.10 in [12], since $\hat{\sigma}$ is adaptive submodular and optimal value is obtained in stage 2, we have

$$\hat{\sigma}(\pi^{\text{greedy}}_{\text{relaxed}}) \geq (1 - e^{-\frac{B_1-1}{B_1}})\hat{\sigma}(\pi^{\text{OPT}}_{\text{relaxed}}).$$

By the definition of $\hat{\sigma}(\pi^{\text{OPT}}_{\text{relaxed}})$, we can see that

$$\hat{\sigma}(\pi^{\text{OPT}}_{\text{relaxed}}) \geq \hat{\sigma}(\pi^{\text{OPT}}).$$

Moreover, from Lemma 5, we have

$$\hat{\sigma}(\pi^{\text{greedy}}_{\text{relaxed}}) = \hat{\sigma}(\pi^{\text{greedy}}).$$

Thus,

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

2.8 Proof of Lemma 6

It has been proven in [1] that the classical influence maximization under independent cascade model is NP-hard. We would like to show that the classical IM can be reduced to our action selection problem. Given an arbitrary instance of the classical IM problem with budget k, the goal is to influence the whole network by initially selecting k nodes. This is only a special case of our problem, where there is only one discount rate $D = \{1\}$ and the action space is $Z = R \times D$. If $L^* \subseteq Z$ is the optimal solution of this action selection problem. Then, we can solve the classical influence maximization problem by selecting corresponding nodes in L^*. It is easy to see that the reduction can be performed within polynomial time. Based on the above analysis, we prove the NP-hardness of the optimal action selection in stage 2.

2.9 Proof of Lemma 7

Suppose the newly reached users are R, and accordingly the action space is $Z = R \times D$. We first prove the submodularity of $Q(L; R|(\lambda, \phi))$, under some realization (λ, ϕ). $\forall E_1, E_2 \subseteq Z$ and $E_1 \subseteq E_2$, $\forall z \in Z \setminus E_2$, we aim to prove that

$$\begin{aligned} &Q(E_1 \cup \{z\}; R|(\lambda, \phi)) - Q(E_1; R|(\lambda, \phi)) \geq \\ &Q(E_2 \cup \{z\}; R|(\lambda, \phi)) - Q(E_2; R|(\lambda, \phi)). \end{aligned} \quad (3)$$

Under the same seeding and diffusion realization (λ, ϕ), $E_1 \subseteq E_2$ implies that the users influenced by actions E_2 are also influenced by actions E_1, but there exist users influenced by E_2 but not influenced by E_1. Therefore, the set of users influenced by z but not influenced by E_1 is a superset of that under E_2. This conclusion directly leads to formula (3). Since the left side of formula (3) represents the number of users influenced by z but not by E_1, and the right side represents the number of users influenced by z but not by E_2, Thus formula (3) holds. Since a non-negative combination of submodular functions is still submodular, we derive that $Q(L; R)$ is submodular.

We next prove that under some realization (λ, ϕ), $Q(L; R|(\lambda, \phi))$ is monotone nondecreasing. $\forall E_1 \subseteq E_2 \subseteq Z$, users who become seeds under E_1 must be seeds under E_2. However, users seeded by $E_2 \setminus E_1$ may become seeds. The states of edges are the same for E_1 and E_2 under the same realization ϕ. Thus, E_2 can achieve at least the same diffusion as E_1, i.e. $Q(E_2; R|(\lambda, \phi)) \geq Q(E_1; R|(\lambda, \phi))$. The non-negative combination of monotone functions are still monotone. Hence, $Q(E_2; R) \geq Q(E_1; R)$.

2.10 Proof of Lemma 8

Under the discrete-discrete setting, the seeding process in stage 1 is the same as the discrete-continuous setting, while the seeding process in stage 2 aims at choosing an optimal subset of actions from Z. Therefore, we only need to modify the proof of Lemma 4 about stage 2. Then, we attempt to prove the inequality $|D'| - |B'| \leq |D| - |B|$. Following the same argument in the proof of Lemma 4, we have $B \subseteq B'$. Suppose the actions adopted in $v(y)$'s neighbors under ψ' are L'. Since $\psi \subseteq \psi'$, by the proof of Lemma 4, we have

$$N(v(y)) \setminus N(\bigcup_{p \in dom(\psi')} v(p)) \subseteq N(v(y)) \setminus N(\bigcup_{p \in dom(\psi)} v(p)). \quad (4)$$

Moreover, the budget brought by $v(y)$ in stage 2 is the same under ψ and ψ'. Thus, we can carry out the same set of actions L' under ψ, triggering the same diffusion from newly reached users. By formula (4), we see that $v(y)$ reaches more new users under ψ than ψ'. Thus, the action space under ψ' is contained in that of ψ, which allows the possibility to achieve a larger diffusion under ψ. Recall $B \subseteq B'$, we have $D' \setminus B' \subseteq D \setminus B$. Moreover, $\hat{\sigma} = |\sigma|$, $B \subseteq D$ and $B' \subseteq D'$. Thus, we obtain $|D'| - |B'| \leq |D| - |B|$. The rest of the proof is the same as the proof of Lemma 4.

2.11 Proof of Theorem 3

Recall that $\hat{\sigma}(\cdot|(\lambda, \phi))$ is adaptive submodular with respect to realization distribution $p((\lambda, \phi))$. From the general result of Theorem A.10 in [27], we see that in each round if we obtain a $Q(L; R)$ of α approximation ratio of the optimal solution, then the greedy policy $\pi^{\text{greedy}}_{\text{discrete}}$ achieves a $1 - e^{-\alpha\frac{B_1-1}{B_1}}$ approximation of the optimal policy $\pi^{\text{OPT}}_{\text{discrete}}$. According to Theorem 1 in [29], since $Q(L; R)$ is monotone and submodular, the approximation ratio of the greedy algorithm in stage 2 is $\frac{1}{2}(1 - e^{-1})$, i.e., $\alpha = \frac{1}{2}(1 - e^{-1})$. Thus, the approximation ratio of $\pi^{\text{greedy}}_{\text{discrete}}$ is $1 - e^{-\frac{B_1-1}{2B_1}(1-\frac{1}{e})}$.

3 Estimation of $I(S)$

Influence estimation is frequently demanded in IM algorithms to decide the discount allocation. However, it is proven to be #P-hard [S1] [S2] and becomes an obstacle of influence maximization. Thanks to the efforts of Tang *et al.* [7] [21], a polling based framework called IMM is proposed for efficient influence estimation, with which the time complexity of greedy algorithm is only $O((k + l)(n + m) \log n/\epsilon^2)$. Due to its favorable performance, we apply the framework to estimate influences when designing allocations, just like many other works, e.g., [8] [9] [S3]. To help readers comprehend the implementation of our experiment, we would like to briefly introduce the framework below.

Definition S1. ***(Reverse Reachable Set) [7] [21]*** *Let v be a node in V. A reverse reachable (RR) set for v is generated by first sampling a graph g from G, and then taking the set of nodes in g that can reach v. A random RR set is an RR set for a node selected uniformly at random from V.*

Lemma S1. ([S4]) *For any seed set S and any node v, the probability that a diffusion process from S can activate v equals the probability that S overlaps an RR set for v.*

Given network $G(V,E)$ and propagation probabilities of each edge, we first derive the transpose graph of G defined as $G^T(V,E^T)$, i.e., edge $(v,u) \in G^T$ iff. $(u,v) \in G$. Note that the propagation probability of edge (v,u) remains to be p_{uv}. The crux of estimating $I(S)$ lies in generating θ random RR sets defined in Definition S1. Briefly speaking, to generate an RR set, we select a node (e.g., v) uniformly at random from V and stimulate a propagation from v. The intuition is that by reverse propagation, we could find out which set of users could potentially influence v. From Lemma S1, it could be inferred that in expectation the influence spread $I(S)$ is equal to the fraction of RR sets covered by S times the number of users $|V|$ [7] [21]. Denoting the number of RR sets covered by seed set S as $F_R(S)$, we have $E[I(S)] = E[F_R(S)|V|/\theta]$.

As can be seen, to estimate the influence spread, the only parameter in need of specification is θ, i.e., the number of RR sets to be generated, which is usually in $O(n \log n)$ [8]. In our experiment, for datasets same with previous works, we check their θ according to the constraints in [7]. If their setting is appropriate, we adopt the same θ as them. Specifically, "wiki-Vote" is adopted in [8] where $\theta = 0.25M$; "com-Dblp" is applied in [8] and [S5], where both works set θ to be 20M; and "soc-Livejournal" is also tested in [S5], where $\theta = 40M$. For the dataset "ca-CondMat" adopted by us, we find the parameter following the process in [7].

To illustrate the setting of θ, we reproduce relevant content in [7] as follows.

According to Theorem 1 in [7], to obtain an approximation of error ϵ with probability $1 - 1/n^l$, θ should be no less than

$$\theta = \frac{2n[(1-1/e)\cdot\alpha + \beta]^2}{\mathrm{OPT}\epsilon^2},$$

where OPT is the maximum expected influence spread,

$$\alpha = \sqrt{l \log n + \log 2}, \text{ and}$$

$$\beta = \sqrt{(1-1/e)\cdot[\log C_n^k + l \log n + \log 2]}.$$

As we know, it is almost impossible to obtain OPT. Thus, we substitute OPT with a lower bound of the influence spread. Accordingly, θ is larger than it is under OPT. Since θ is expected to be large, the estimation is even more precise after substitution. With this idea, we check the setting of previous works which adopt identical datasets and find their values of θ are reasonable. Regarding the dataset "ca-CondMat", we consider the most demanding case $k = 50$ and set ϵ to be 0.05 for rigor. The resultant θ is 1.4M. For safety, we set θ to be 2M as shown in the main paper.

The above basic technique could apply to the adaptive case well, where we only need to estimate the influence of a deterministic seed set (in each round the seeded user will certainly become the seed or not, accepting or refusing the discount). However, for the non-adaptive case, given an allocation, since no observation is made, the state of users is probabilistic and accordingly the influence spread is related to $p_u(c_u)$. When estimating the influence, we need to apply Theorem 9 in [8]. Specifically, when computing $\hat{f}(C_1;X)$ via Equation (10), we need to estimate the value of $\max \hat{Q}(C_2;\cdot)$. By running Alg. 1, we could optimize $\hat{Q}(C_2;\cdot)$ and obtain the corresponding allocation (assumed to be C). Then, by Theorem 9 in [8], we know that the expected influence is $n\cdot[\sum_{h\in \text{RR sets}} 1 - \prod_{u\in h}(1-p_u(c_u))]/\theta$. Since $p_u(c_u)$ is available, the value of the expression could be derived. Accordingly, the value of $\hat{f}(C_1;X)$ could be derived by Eq. (10).

We would also like to mention that the IMM method is only to estimate the influence when designing allocations. However, when experimentally testing the performance of algorithms, the influence spread is obtained by running 20K times Monte Carlo simulations, as indicated in Sec. 7.1.

4 More Experimental Results

In this section, we report the influence spread (Fig. 3, Fig. 4) and running time (Fig. 5, Fig. 6) under $\alpha = 0.8$.

5 References

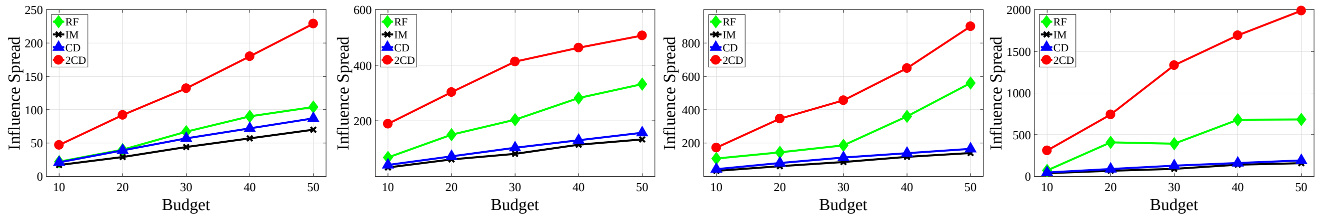
[S1] W. Chen, C. Wang, and Y. Wang, Scalable influence maximization for prevalent viral marketing in large-scale social networks. in *Proc. Int. Conf. Knowl. Discov. Data Min., KDD*, ACM, 2010, pp. 1029–1038.

[S2] 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.

[S3] K. Huang, J. Tang, X. Xiao, A. Sun, and A. Lim. Efficient Approximation Algorithms for Adaptive Target Profit Maximization, in *Proc. Int. Conf. Data Eng. ICDE*, IEEE, 2020, to appear.

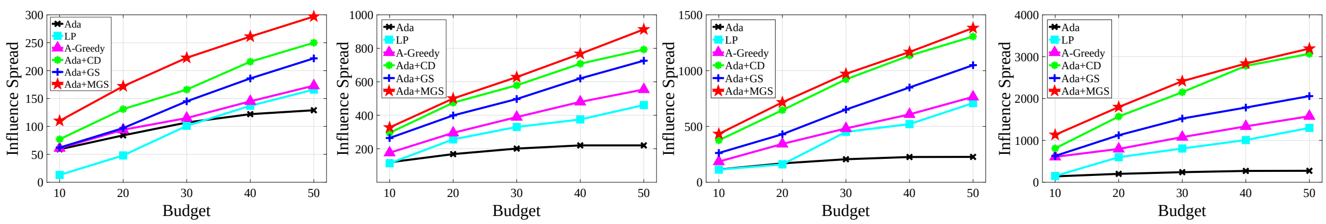
[S4] C. Borgs, M. Brautbar, J. Chayes, and B. Lucier. Maximizing social influence in nearly optimal time. in *Proc. Annu. ACM-SIAM Symp. Discrete Algorithms*, 2014, pp. 946–957.

[S5] J. Yuan and S. Tang. Adaptive Discount Allocation in Social Networks. https://www.dropbox.com/s/tlm5ldll0md6lu2/TRmobihoc.pdf?dl=0. (2017).



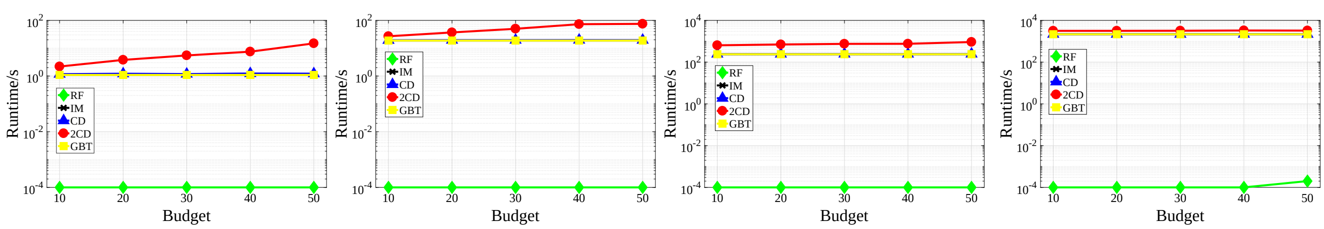
(a) Wiki-Vote (b) Ca-CondMat (c) com-Dblp (d) soc-Livejournal

Fig. 3. Influence Spread in the Non-adaptive Case (α=0.8).



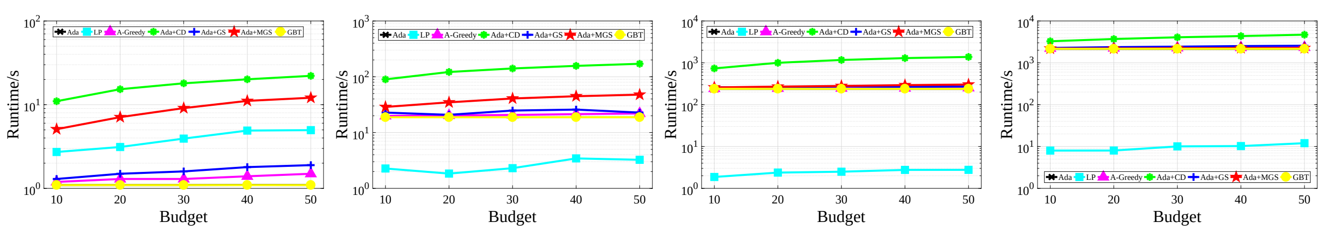
(a) Wiki-Vote (b) Ca-CondMat (c) com-Dblp (d) soc-Livejournal

Fig. 4. Influence Spread in the Adaptive Case (α=0.8).



(a) Wiki-Vote (b) Ca-CondMat (c) com-Dblp (d) soc-Livejournal

Fig. 5. Running Time in the Non-adaptive Case (α=0.8).



(a) Wiki-Vote (b) Ca-CondMat (c) com-Dblp (d) soc-Livejournal

Fig. 6. Running Time in the Adaptive Case (α=0.8).