Efficient Algorithms for Budgeted Influence Maximization on Massive Social Networks

Song Bian Qintian Guo Sibo Wang Jeffrey Xu Yu
The Chinese University of Hong Kong
{sbian,qtguo,swang,yu}@se.cuhk.edu.hk

ABSTRACT

Given a social network G, a cost associated with each node, and a budget B, the *budgeted influence maximization (BIM)* problem aims to find a set S of nodes, denoted as the seed set, that maximizes the expected number of influenced users under the constraint that the total cost of the users in S is no larger than B. The current state-of-the-art practical solution for BIM problem provides a $(\frac{1-1/e}{2}-\epsilon)$-approximate ($\approx 0.316-\epsilon$) result and is still inefficient on large networks. We first show that we can improve the approximation guarantee to $1-1/e^{\beta}-\epsilon$ where $1-1/e^{\beta}=(1-\beta)(1-1/e)$, achieving a better approximation guarantee ($\approx 0.355-\epsilon$).

Next, we apply the reverse sampling based technique, a popular technique for classic influence maximization, to our studied BIM problem. However, it is non-trivial to design efficient solutions for large scale networks even the reverse sampling based technique is applied. On one hand, it is unclear how to derive tight bounds for the nodes selected by the greedy algorithm under the budgeted scenario, where each time it selects the seed node with the highest benefit-cost ratio. With tighter bounds, the algorithm can terminate as soon as the approximation ratio is satisfied, thus saving the running cost. On the other hand, the number of nodes selected under BIM problem may be quite large since it may greedily select many nodes with large benefit-cost ratio but with low costs. The time complexity of existing influence maximization algorithms heavily depends on the size of the seed set. To tackle such challenging issues, we first present new bound estimation techniques for the BIM problem. Next, we present new node selection strategies to alleviate the dependency to the size of the seed set. Extensive experiments show that our proposed solution is far more efficient than alternatives.

PVLDB Reference Format:
Song Bian, Qintian Guo, Sibo Wang, Jeffrey Xu Yu. Efficient Algorithms for Budgeted Influence Maximization on Massive Social Networks. *PVLDB*, 13(9): 1498-1510, 2020.
DOI: https://doi.org/10.14778/3397230.3397244

Proceedings of the VLDB Endowment, Vol. 13, No. 9
ISSN 2150-8097.
DOI: https://doi.org/10.14778/3397230.3397244

1. INTRODUCTION

The *influence maximization (IM)* problem has been studied for decades due to its important application in viral marketing [24], rumor monitoring [11], and outbreak detection [28]. Given a graph G and an integer k, the influence maximization problem aims to find a set S of nodes, denoted as the seed set, that maximizes the expected influence among all size-k seed sets. In the classic setting, all nodes are assumed to have equal cost and the cost to invite the k nodes is ignored. While in reality, the nodes in social networks typically have different costs. For instance, based on the recent report on news and social media [1, 2, 3, 4, 5, 6], it usually costs more to invite influential users, e.g. Rihanna, than to invite normal users to do the advertisement.

In this paper, we consider the *budgeted influence maximization (BIM)* problem, where each node is associated with a cost and a budget B is taken as the input. The goal is to find a seed set S that achieves the highest expected influence under the constraint that the sum of the costs of each node in S does not exceed the budget. Notice that the classic IM problem is a special case of the BIM problem where the cost of each node is the same. Since the IM problem is NP-hard [24], it is easy to verify that the BIM problem is also NP-hard. Therefore, a line of research works focus on developing heuristic algorithms, e.g., [30, 23], to reduce the computational costs. Such heuristics, however, provide no guarantee on the returned answer.

To remedy this deficiency, Khuller et al. [25] first present a $(1-1/e)$-approximate solution which requires enumerating all the size-3 seed set and apply the greedy algorithm to select the remaining nodes. Since there are $O(n^3)$ different size-3 seed set, it incurs prohibitive computational cost and is impractical on large social networks. To overcome the high computational cost, they further present a $(\frac{1-1/e}{2})$-approximate solution by applying a modified greedy strategy. However, the previous solutions all assume that we can effectively calculate the expected influence of a seed set S, which is #P-hard in general. To tackle this issue, Nguyen et al. [31] present a framework that applies the reverse sampling technique to effectively provide an estimation of the expected influence of a seed set S, and returns a $(\frac{1-1/e}{2}-\epsilon)$-approximate answer. There also exist some theoretical studies, e.g., [18], that try to return a $(1-1/e)$-approximate solution with reduced time complexity. However, such solutions are still of only theoretical interest and are in general impractical to real scenarios. Therefore, to our knowledge, the most practical approximate solution is still the one proposed by Nguyen et al. [31], which simply applied the reverse sampling technique that is widely used in the influence maximization literature. However, as will be shown in our experimental study, the proposed solution in [31] still incurs unnecessarily high computational cost since they do not consider the properties of the BIM problem.

Motivated by this, we present a new framework *IMAGE*[1] to devise more efficient approximate algorithms to the BIM problem. Our framework first includes a new bound estimation scheme for the BIM problem, which allows the algorithm to terminate as soon as the approximation guarantee is satisfied. Besides, under the BIM setting, the number of selected nodes might be quite large since quite a lot of nodes with low costs might be selected. The computational cost of existing algorithms heavily depends on the size of the seed set returned. We show an effective node selection strategy that alleviates the dependency on the size of the seed set. We further demonstrate that the time complexity of our proposed algorithm only depends on the budget B and the input graph size, not the seed set size. In summary, we make the following contributions:

- We propose an effective bound estimation scheme for the BIM problem and it allows the algorithm to terminate as soon as the approximation ratio is satisfied;
- We present an effective greedy strategy to alleviate the dependency to the size of the seed set and show that our algorithm still provides strong approximation guarantee;
- We present theoretical analysis on the time complexity of our proposed algorithm. We show that the expected running time of our proposed algorithm depends on the budget B and the input graph size, not depending on the size of the seed set returned.
- We conduct extensive experiments to evaluate the performance of our algorithms against alternatives on large social graphs with up to 41.7 million nodes. Our experimental results show that our proposed solution is far more efficient than alternatives.

2. PRELIMINARIES

We present problem definition and necessary background in Section 2.1 and revisit existing solutions for the problem in Section 2.2. Table 1 shows the frequently used notations.

2.1 Problem Definition

We abstract a social network as a directed graph $G = (V, E)$, where V is the set of users and E is the set of edges such that each edge indicates the friendship/followed-by relationship between users. We denote n as the number of nodes in G and m as the number of edges in G. Besides, for any two nodes u and v in the social network, if there exists an edge $\langle u, v\rangle$, we say that u is an incoming neighbor of v, and v is an outgoing neighbor of u. For each directed edge $e = \langle u, v\rangle$, it is associated with a propagation probability $p(e) \in [0, 1]$. Given the input graph G and seed set S, the influence propagation follows a certain cascade model $\mathbb{C}$.

In this paper, we focus on two cascade models: the *independent cascade (IC)* model and *linear threshold (LT)* model, which are the most widely used models in the literature. At the beginning of the influence propagation, the seed set S becomes activated, and other nodes are not activated. When a node is activated at timestamp i, it has only one chance to activate its out-neighbors which are inactivated at timestamp $i + 1$. After that, the node is not able to activate other nodes anymore. The main difference between the two models lies in how an inactive node gets activated.

Table 1: Notation Table

Notation	**Description**
$G(V, E)$	a social network
n, m	the number of nodes and number of edges in G
B	the total budget
$c(S)$	the total cost of the seed set S
$\sigma(S)$	the expected spread of seed set S
$\mathcal{R}$	a set of random RR sets
$\Lambda(S)$	the number of RR sets in $\mathcal{R}$ that is covered by S
$\Lambda(v\|S)$	the number of random RR sets in $\mathcal{R}$ that is covered by v but not by S
S'_{opt}	the seed set that maximizes the coverage on $\mathcal{R}$ under budget B
τ_B	The number of sets satisfying that such a set has cost no more than B but adding any other node to the set will make the cost larger than B
S_{opt}	the optimal seed set for BIM

- **In IC model**, if a node u is activated at timestamp i, then at timestamp $(i + 1)$, u is able to activate v which is an out-neighbor of u with a probability $p(u, v)$.
- **In LT model**, there is an activation threshold λ_v which is a parameter uniformly and randomly selected in range $[0, 1]$. If a node v is inactive at timestamp i, node v gets activated at timestamp $i+1$ if and only if $\sum_{u\in N} p(u, v) \geq \lambda_v$, where N is the set of all the in-neighbors of v that is activated by timestamp i.

Let $\sigma_{\mathbb{C}}(S)$ be the expected number of nodes activated by a seed set S during the influence propagation under cascade model $\mathbb{C}$, referred to as the expected influence of S. We omit the subscript $\mathbb{C}$ and use $\sigma(S)$ to denote the expected influence if the context is clear. The IM problem asks for a size-k seed set S that maximizes the expected influence. However, as we mentioned in Section 1, the cost to invite a user to do advertisements for a product usually differs a lot and it is sometimes impractical to ignore such cost differences. Hence, we study the budgeted influence maximization problem, which takes different costs into account.

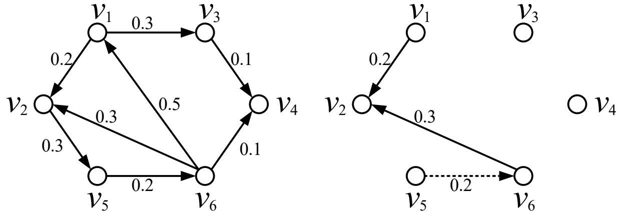
Definition 1. *(Budgeted Influence Maximization). Let $G = (V, E)$ be the input graph where each edge $e \in E$ is associated with a probability $p(e)$ and each node $v \in V$ is associated with a cost $c(v)$. Given a budget B and a cascade model $\mathbb{C}$, the goal of the budgeted influence maximization is to find the seed set S that gains the largest expected influence under $\mathbb{C}$ while satisfying $\sum_{s\in S} c(s) \leq B$.*

2.2 Existing Solutions Revisited

Classic Influence Maximization. In the literature, most existing approximate algorithms for IM rely on the *reverse sampling* based technique proposed by Borgs et al. [10]. This technique is based on the concept of *random reverse reachable (RR)* sets. A random RR set is generated in two steps:

- Select a node v from V uniformly at random;

[1] Budgeted Influence Maximization with threshold Greedy and Bound Refinement



(a) Social network G (b) An RR set sample

Figure 1: An example of the RR set construction.

- Produce a sample set R of the nodes in V, such that the probability a node u appears in R is equal to the probability that u can activate node v.

As shown in previous work [37, 36, 35], a random RR set can be efficiently generated for both the IC model and LT model with a stochastic BFS traversal from a randomly selected node following the reverse direction of the edges in graph G. Below shows an example of the RR set construction under the IC model. We refer readers to [35] for how random RR sets are generated under the LT model.

Example 1. Consider the social network G in Figure 1(a). The number on each edge denotes the propagation probability of the edge. A random RR set sample in Figure 1(b) is constructed as follows. Firstly, node v_2 is sampled uniformly at random from G, leading to $R = \{v_2\}$ at this moment. Then we perform a stochastic BFS from v_2, following the reverse direction of incoming edges. We generate random numbers $r_1 = 0.1$ and $r_2 = 0.2$ for the incoming edges of node v_2, (v_1, v_2) and (v_6, v_2), respectively. Since $r_1 < 0.2$ and $r_2 < 0.3$, nodes v_1 and v_6 are activated by v_2, and are put into R; that is, $R = \{v_2, v_1, v_6\}$. Since node v_1 has only one in-neighbor v_6 and $v_6 \in R$, we do nothing for node v_1. Then we generate a random number $r_3 = 0.4$ for the incoming edge (v_5, v_6) of node v_6. Node v_5 is not activated and we do not add v_5 into R since $r_3 > 0.2$. In Figure 1(b), the dashed line between node v_5 and v_6 indicates this traversal failure. At this time, no more nodes can be traversed, so the stochastic BFS terminates with $R = \{v_2, v_1, v_6\}$. □

Borgs et al. [10] establish the following lemma to use random RR sets to estimate the expected influence of a seed set S.

Lemma 1. *Assume that θ random RR sets are generated. Let x_i ($i \in [1, \theta]$) be a random variable to be 1 if $S \cap R_i \neq \emptyset$ and 0 otherwise. Let $\sigma(S)$ be the expected influence of seed set S. The following equation holds.*

$$\sigma(S) = \frac{n}{\theta} \cdot \mathbb{E}[\sum_{i=1}^{\theta} x_i] = \frac{n}{\theta} \sum_{i=1}^{\theta} \mathbb{E}[x_i] \qquad (1)$$

Given θ samples of the RR sets, we say a set S covers a RR set R if $S \cap R \neq \emptyset$ and denote $\Lambda(S)$ as the number of RR sets that are covered by S in the θ RR sets. Then the above lemma indicates that the expected influence of S is linearly dependent on the fraction of random RR sets covered by S. Based on the above equation, Borg et al. [10] propose to: *(i)* sample a sufficient number θ of random RR set; *(ii)* apply a greedy algorithm to select k nodes such that in each iteration we select the node that covers the maximum number of RR sets that are not covered by previously selected nodes. The pseudo-code of the greedy algorithm is shown in Algorithm 1. Note that $\Lambda(v|S)$ indicates the number of RR set covered by $S \cup \{v\}$ but not covered by S., i.e.,

$$\Lambda(v|S) = \Lambda(S \cup \{v\}) - \Lambda(S).$$

Algorithm 1: Maximum-Coverage Greedy

```
Input: An integer $k$ and a set $\mathcal{R}$ of random RR sets
Output: A seed set $S$
$S \leftarrow \emptyset$;
for $i$ from 1 to $k$ do
    $u = \arg\max_{v \in V \setminus S} \Lambda(v|S)$;
    $S \leftarrow S \cup \{u\}$;
return $S$;
```

Borgs et al. prove that by utilizing the simple greedy algorithm, one can obtain a seed set S which provides a $(1 - 1/e - \epsilon)$-approximate solution for the classic influence maximization problem with $1 - \delta$ probablity with a time complexity of $O(k \cdot (n + m) \cdot \ln^2(n) \cdot \epsilon^{-3})$. Later, Tang et al. [37, 36] propose TIM/TIM+ and IMM that improves the performance over Borgs et al.'s method and reduce the time complexity to $O(k \cdot (n + m) \cdot \ln(n) \cdot \epsilon^{-2})$. Nguyen et al. [32] present SSA/DSSA and Tang et al. [35] present OPIM-C to further improve the practical performance but retain the same expected time complexity.

Budgeted Influence Maximization. However, all the above solutions are designed for the classic influence maximization and do not take the cost into consideration. To apply to the budgeted influence maximization problem, a natural idea is to modify the greedy algorithm so that each time it greedily selects the node that maximizes the benefit-cost ratio, i.e., the node whose gain on the expected influence (given previously selected nodes) over its cost is maximized. However, such an extension provides no constant approximation guarantee. We explain with the following example.

Example 2. Given a social network with a node set $V = \{u, v_1, v_2, \cdots, v_{n-1}\}$, where node u is an isolated node, while nodes $v_1, v_2, \cdots, v_{n-1}$ are fully connected with edges whose propagation probability all equal 1. The cost $c(u) = 1 - \epsilon$ and $c(v_i) = n-1$ for $1 \leq i \leq n-1$. Given a budget $B = n-1$, the optimal solution for this problem is to choose an arbitrary node from $\{v_1, v_2, \cdots, v_{n-1}\}$. However, if we greedily select the node with the maximum benefit-cost ratio, node u will be selected first, since it has the largest benefit-cost ratio $1/(1 - \epsilon)$ while all the other nodes have a benefit-cost ratio of 1. After taking node u, no more node can be added since otherwise it will exceed the budget. Then, the approximation ratio for this algorithm is $1/(n - 1)$ and the approximation ratio can be arbitrarily bad as n grows. □

The greedy strategy which iteratively selects the node with the maximum benefit-cost ratio provides no constant approximation guarantee but by a slight modification, it can help obtain a constant approximation guarantee. Since it is #P-hard to derive the exact expected influence, we follow the paradigm in classic influence maximization and consider the coverage $\Lambda(S)$ of a set S on a set $\mathcal{R}$ of random RR sets, whose expectation is exactly the expected influence $\sigma(S)$ of a set S. The modified version, dubbed as budgeted greedy, is proposed by Khuller et al. [25]. The pseudo-code (tailed

Algorithm 2: Budgeted Maximum-Coverage Greedy

```
  Input: $G = (V, E)$, budget $B$, a set $\mathcal{R}$ of RR sets
  Output: A seed set $S$
1 $S \leftarrow \emptyset$, $V' \leftarrow V$;
2 while $V' \neq \emptyset$ do
3     $u \leftarrow \arg\max_{v \in V'} (\Lambda(v|S)/c(v))$;
4     if $c(S \cup \{u\}) \leq B$ then
5         $S \leftarrow S \cup \{u\}$;
6     $V' \leftarrow V' \setminus \{u\}$;
7 $s \leftarrow \arg\max_{v \in V, c(v) \leq B} \Lambda(\{v\})$;
8 return $\arg\max(\Lambda(\{s\}), \Lambda(S))$;
```

Algorithm 3: IMAGE(G, B, δ)

```
1  Initialize $\theta_0$ and $\theta_{max}$;
2  $\theta \leftarrow \theta_0$, $i_{max} \leftarrow \log_2 \theta_{max}/\theta_0$;
3  while $\theta \leq \theta_{max}$ do
4      sample $\mathcal{R}_1$ and $\mathcal{R}_2$, each with $\theta$ random RR sets;
5      select the seed node $S$ with $\mathcal{R}_1$ with certain
         greedy strategy, e.g., budgeted greedy;
6      verify if $S$ provides approximation ratio on $\mathcal{R}_2$
         with at least $1 - \frac{2\delta}{3 i_{max}}$ probability;
7      if $S$ provides desired approximation ratio then
8          return $S$;
9      $\theta \leftarrow 2\theta$;
10 return $S$;
```

for BIM) is shown in Algorithm 2. The main idea is to generate two sets of seed set. The first seed set S is generated by greedily selecting the node with the maximum benefit-cost ratio, i.e., the node whose marginal coverage $\Lambda(v|S)$ over its cost $c(v)$ is maximum, until adding a new node will make the cost of the seed set exceed the budget (Algorithm 2 Lines 3-6). The other seed set contains only a single node, which is the node that has the largest coverage on $\mathcal{R}$ among all nodes whose cost is no larger than B (Algorithm 2 Line 7). Finally, it selects the seed set that has larger expected influence (Algorithm 2 Line 8). The following example is given to illustrate Algorithm 2.

EXAMPLE 3. Consider the social network G in Figure 1(a). Given buddget $B = 3$ and assume that the cost of the nodes are $c(v_1) = 1, c(v_2) = 2, c(v_3) = 2, c(v_4) = 1, c(v_5) = 2$, and $c(v_6) = 2$, respectively. We then sample a set $\mathcal{R}$ of RR sets, and assume that $\mathcal{R} = \{\{v_2, v_6\}, \{v_1, v_6\}, \{v_3\}, \{v_5, v_2\}, \{v_3, v_1, v_6\}\}$. According to Algorithm 2 Lines 2-6, the first node we select is v_1 since v_1 has the maximum benefit-cost ratio, $2/1 = 2$. After node v_1 is selected, the marginal benefit-cost ratio of v_2, v_3, v_4, v_5 and v_6 becomes $2/2, 1/2, 0/1, 1/2$ and $1/2$, respectively. Thus, we select v_2 as the second seed. Notice that at this moment, $c(S) = c(v_1) + c(v_2) = 3$. It reaches the budget B, and no more node will be added into S. So the first solution is $S = \{v_1, v_2\}$. Then according to Algorithm 2 Line 7, we obtain the second solution $\{v_6\}$, since node v_6 has the maximum benefit $\Lambda(\{v_6\}) = 3$ among all nodes. Because $\Lambda(\{v_1, v_2\}) > \Lambda(\{v_6\})$, the final output of Algorithm 2 is $\{v_1, v_2\}$. □

Khuller et al. [25] show that the budgeted greedy, i.e., Algorithm 2, provides a $(\frac{1-1/e}{2})$-approximate solution for maximizing the coverage under budget B, and further demonstrate that a tighter bound of $(1 - 1/\sqrt{e})$ can be achieved for Algorithm 2. However, Zhang et al. [39] pinpoint out that the proof of the $(1 - 1/\sqrt{e})$-approximation is problematic. Nguyen et al. [31] present a seminal work by combing the RR-set based technique and the budgeted greedy to derive a $(1 - 1/\sqrt{e} - \epsilon)$-approximate solution. Their proof contains the same loopholes as that in [25] and therefore only provides a $\frac{1-1/e}{2} - \epsilon$ approximation guarantee. The main idea is to sample a sufficient number of random RR sets and then apply the budgeted greedy algorithm based on the coverage of the seed nodes on the RR sets. However, their solution still leaves much room for improvement in terms of running time since they do not take into account the properties of the BIM problem. This motivates us to propose our IMAGE framework, detailed as follows.

3. PROPOSED FRAMEWORK

3.1 Overview

At a high level, our proposed IMAGE shares a similar framework of existing approximate IM algorithms [35, 32] in that our IMAGE also runs in an iterative manner and in each iteration, it includes three phases:

1. **RR set generation phase.** In this phase, we sample two sets, denoted as $\mathcal{R}_1$ and $\mathcal{R}_2$ of random RR sets, both containing θ random RR sets.
2. **Seed set selection phase.** In this phase, we use $\mathcal{R}_1$ to select seed nodes. For instance, we may apply the budgeted greedy algorithm on $\mathcal{R}_1$ to select the seed nodes.
3. **Seed set verification phase.** In this phase, we use $\mathcal{R}_2$ to verify if the seed set selected in the previous phase provides an approximation guarantee. If the seed set can provide the approximation ratio as desired, the seed set is returned. Otherwise, the number of RR sets in $\mathcal{R}_1$ and $\mathcal{R}_2$ is doubled to $2 \cdot \theta$ and we repeat the above process.

Algorithm 3 shows the pseudo-code of our IMAGE framework. We will elaborate on how we select θ_0 and θ_{max} later in Section 4. Currently, we assume that these two parameters are given. In [31], Nguyen et al. simply apply the budgeted greedy to select the seed set S (Algorithm 3 Line 5). To check if the algorithm provides $(\frac{1-1/e}{2} - \epsilon)$-approximate solution, it simply checks if the RR set covered by the seed set S selected exceeds certain thresholds depending on ϵ, which is rather conservative. Both strategies are ineffective and result in unnecessarily high computational costs. Next, we present our first algorithm by deriving new bounds based on the seed sets selected with the budgeted greedy algorithm so as to reduce the computational cost.

3.2 Tightened Bounds of BIM

According to Equation 1, a random RR set provides an unbiased estimation for the expected influence of an arbitrary seed set S. Therefore, we can first sample a sufficient number of random RR set and then apply concentration bound to derive the estimation errors of $\sigma(S)$. We use the following concentration bounds tailed for influence maximization.

Lemma 2 ([35, 36]). *Given a set $\mathcal{R}$ of θ random RR sets and a seed set S, let $\Lambda(S)$ stands for the coverage of S on $\mathcal{R}$. Given $\eta \geq 0$, the following inequalities hold:*

$$\Pr[\Lambda(S) - \sigma(S) \cdot \frac{\theta}{n} \geq \eta] \leq \exp(-\frac{\eta^2}{2\sigma(S) \cdot \frac{\theta}{n} + \frac{2}{3}\eta})$$

$$\Pr[\Lambda(S) - \sigma(S) \cdot \frac{\theta}{n} \leq -\eta] \leq \exp(-\frac{\eta^2}{2\sigma(S) \cdot \frac{\theta}{n}}).$$

Given Lemma 2, we can provide an upper bound and a lower bound for any fixed set S. Therefore, we can first apply the budgeted greedy on $\mathcal{R}_1$ to find the seed set S. According to [25, 31], we can derive a $(\frac{1-1/e}{2})$-approximate solution in terms of the coverage on $\mathcal{R}_1$ under budget B. That is to say, let S_g be the seed set selected by the budgeted greedy algorithm, it guarantees that $\Lambda(S) \geq (\frac{1-1/e}{2}) \cdot \Lambda(S'_{opt})$, where S'_{opt} is the seed set that maximizes the coverage on $\mathcal{R}_1$. Let S_{opt} be the optimal seed set that maximizes the expected influence. Then, we have that:

$$\Lambda(S_g) \geq \frac{1-1/e}{2} \cdot \Lambda(S'_{opt}) \geq \frac{1-1/e}{2} \cdot \Lambda(S_{opt})$$

By applying Lemma 2, we can derive an upper bound $\sigma^+(S_{opt})$ of $\sigma(S_{opt})$ according to $\Lambda(S_{opt})$. With $\mathcal{R}_2$, we can further derive a lower bound $\sigma^-(S_g)$ of $\sigma(S_g)$. We did not use $\mathcal{R}_1$ to derive the upper bound of $\sigma(S_g)$ since there is dependency between $\mathcal{R}_1$ and S_g, i.e., S_g is selected based on $\mathcal{R}_1$. The algorithm terminates as soon as $\sigma^-(S_g)/\sigma^+(S_{opt})$ is larger than the approximation ratio required.

However, such an upper bound is still very loose in two aspects. Firstly, the current worst-case approximation ratio $\frac{1-1/e}{2}$ of the budgeted greedy algorithm is actually still quite loose. Secondly, the actual approximation ratio is usually much better than the worst-case scenario. If we can derive a more accurate upper bound, then the algorithm can terminate earlier and save running costs.

Improved worst-case bound. We show that we can improve the worst-case approximation ratio from $(\frac{1-1/e}{2})$ (≈ 0.316) to $(1-1/e^{\beta})$ where β satisfies that $(1-\beta)(1-1/e) = 1-1/e^{\beta}$, in which case $(1-1/e^{\beta}) \approx 0.355$. Let S_g be the seed set selected by the budgeted greedy algorithm (Algorithm 2 Lines 2-6). Denote S_i as the set of seed nodes selected by the first i iterations *without skipping any node*. The following lemma[2] holds for S_i.

Lemma 3 ([25, 31]). *Let v_i be the node selected in the i-th iteration without skipping any nodes, i.e., Algorithm 2 Line 4 holds for all i iterations, and S_i be the set of nodes selected in the first i iterations ($i = 1, 2, 3, \cdots$) by Algorithm 2 Lines 3-6, the following inequality holds:*

$$\Lambda(S_i) \geq [1 - \prod_{k=1}^{i}(1 - \frac{c_k}{B})] \cdot \Lambda(S'_{opt})$$

According to Lemma 3, we prove that the budgeted greedy algorithm proposed by Khuller et al. [25] can achieve a $(1-1/e^{\beta})$-approximate solution, where $(1-\beta)(1-1/e) = 1-1/e^{\beta}$ and $1-1/e^{\beta} \approx 0.355$. Let S_g be the set returned by budgeted greedy, S'_{opt} be the set maximizing the coverage on $\mathcal{R}_1$ under budget B, and S_{opt} be the set maximizing the expected influence under budget B. We have Theorem 1.

Theorem 1. *The budgeted greedy returns a $(1-1/e^{\beta})$-approximate solution with $(1-\beta)(1-1/e) = 1-1/e^{\beta}$, i.e.,*

$$\Lambda(S_g) \geq (1-1/e^{\beta})\Lambda(S'_{opt}) \geq (1-1/e^{\beta})\Lambda(S_{opt}).$$

With Theorem 1 and Lemma 2, we can derive a tighter upper bound σ^+_{opt} for $\sigma(S_{opt})$ as follows. We use $\frac{\Lambda(S_g)}{(1-1/e^{\beta})}$ as an upper bound for $\Lambda(S_{opt})$. Then, we further apply Lemma 2 to derive an upper bound $\sigma^+_{opt} = \frac{\Lambda(S_g)}{(1-1/e^{\beta})} + \eta$ of $\sigma(S_{opt})$. However, this upper bound is still loose in practice since it is the worst-case guarantee and in practice it is rare for such worst-case bound occurs. In the literature, some existing works focus on improving the upper bound [28, 35] for classic influence maximization problems. However, all these bounds cannot be applied to our studied problem since they do not take into account the impact of cost differences. This motivates us to derive new upper bounds for BIM.

Tightened upper bound. Recap that $\Lambda(v|S)$ indicates the number of RR sets that are covered by v but not covered by S. Given an arbitrary set S, define u_i as the node that has the i-th largest benefit-cost ratio with respect to S, i.e., $\Lambda(u_i|S)$ is the i-th largest. Let S'_{opt} be the seed set that maximizes the coverage on $\mathcal{R}$. We further define S''_{opt} as the seed set that maximizes the coverage on $\mathcal{R}$ but allows containing some fraction of a node. Therefore, the difference between the S'_{opt} and S''_{opt} is that S''_{opt} could contain at most one broken node among all nodes, while S'_{opt} only includes integral nodes. The broken node means that if the cost of the node v is $c(v)$, when we pay the node budget b, the influence it will return is $\Lambda(v) \cdot b/c(v)$. We further define $\kappa(S, B)$ as the set that includes $u_1, u_2, \cdots, u_{i-1}$, and part of the u_i which makes the budget equal to B. We have the following lemma to upper bound the $\Lambda(S_{opt})$, where S_{opt} is optimal solution under budget B for BIM problem.

Lemma 4. *For any seed set S, we have that:*

$$\Lambda(S_{opt}) \leq \Lambda(S'_{opt}) \leq \Lambda(S''_{opt}) \leq \Lambda(S) + \sum_{v \in \kappa(S,B)} \Lambda(v|S)$$

Proof. $\Lambda(S_{opt}) \leq \Lambda(S'_{opt}) \leq \Lambda(S''_{opt})$ naturally holds due to their definitions. Next we prove the second part.

$$\Lambda(S''_{opt}) \leq \Lambda(S''_{opt} \cup S) \leq \Lambda(S) + \sum_{v \in S''_{opt} \setminus S} \Lambda(v|S)$$

$$\leq \Lambda(S) + \sum_{v \in \kappa(S, c(S''_{opt} \setminus S))} \Lambda(v|S) \leq \Lambda(S) + \sum_{v \in \kappa(S,B)} \Lambda(v|S)$$

The third equality is due to the following reason: if the set is allowed to contain at most one broken node, then, we assume that we have a set $S = \{v_1, v_2, \cdots, v_k\}$ which is ordered by the benefit-cost ratio of each node from large to small. We also assume that $S''_{opt} \setminus S = \{v_{o1}, v_{o2}, \cdots, v_{ok}\}$ which is also ordered by the benefit-cost ratio of each node from large to small. Then, if for any $0 \leq i \leq k$, $v_i = v_{oi}$, we could get that the third inequality holds. While if there exists $v_i > v_{oi}$, then we could replace v_{oi} with v_i to obtain a better solution than $S''_{opt} \setminus S$. Therefore, $\kappa(S, B)$ achieves a larger coverage than $S''_{opt} \setminus S$, and this indicates why the third inequality holds, which finishes the proof. □

Let S_i be the set of nodes selected in the first i iterations by Algorithm 2 Lines 2-6. Let k be the number of nodes

[2] Omitted proofs can be found in appendix.

selected by Algorithm 2 Lines 2-6. Combining with Lemma 4, we have a new upper bound of $\Lambda(S_{opt})$ as follows:

$$\Lambda_1^+(S_{opt}) = \min_{0 \leq i \leq k}(\Lambda(S_i) + \sum_{v \in \kappa(S_i, B)} \Lambda(v|S_i))$$

Then, we use $\min(\Lambda(S_g)/(1-e^{-\beta}), \Lambda_1^+(S_{opt}))$ as the final upper bound, which is in practice usually tighter than the worst-case bound. Despite effective the new bounds are, the budgeted greedy is still inefficient since it may include many iterations, and the running cost linearly depends on the number of seed selected. Next, we present our new seed selection strategy to alleviate such dependencies while still providing an approximation guarantee.

3.3 Efficient Node Selection

In the budgeted influence maximization, one big challenge is that given budget B, we may select many nodes with low costs but with a high benefit-cost ratio. However, the time complexity of existing RR-set based solutions heavily depends on the size of the seed set. On one hand, the number of RR set samples in the worst case scenario depends on the size of the seed set. This can be alleviated by being optimistic about the seed set selected as shown in [35, 32].

Another issue is that the time complexity of budgeted greedy linearly depends on the size of the seed set selected by Algorithm 2 Lines 2-6. This differs from the classic IM problems, where we can maintain an inverted index and the time complexity linearly depends on the input size of the RR sets. To explain, the budgeted greedy algorithm needs to repeatedly select the node with the maximum benefit-cost ratio that is usually non-integers, which are more challenging than maximizing the coverage that is integers.

To tackle this challenging issue, we borrow the idea of the threshold-greedy algorithm presented in [9] which was only designed for the case without budget. We extend the idea to budgeted case and prove that we can still provide an approximate solution with a strong theoretical guarantee.

Algorithm details and optimizations. The overall framework of our proposed new algorithm, *budgeted threshold-greedy*, shares a similar framework of the budgeted greedy algorithm in that they both find two sets of seed set and one set includes only a single node which has the maximum coverage whose cost is no larger than B. The main difference lies in Algorithm 4 Lines 2-9. In particular, we first find the node with the maximum benefit-cost ratio (Algorithm 4 Line 2). Denote this maximum benefit-cost-ratio as d_{max}. Then, the algorithm runs in an iterative manner. In the i-th iteration, it uses $d \cdot (1-\xi)^i$ as the threshold, scans all the remaining node, and add a node v to S if $\Lambda(v|S)/c(v)$ is no smaller than the threshold $d \cdot (1-\xi)^i$ (Algorithm 4 Lines 5-7). This allows us to add more than one node in a single iteration, which relaxes the dependency to k. The algorithm terminates either when we cannot add any node to S or when the benefit-cost ratio is below $1/\max_{v \in V} c(v)$. To explain, the coverage of a node is at least 1, otherwise, it is meaningless to add this node. Therefore, the min benefit-cost ratio is at least $1/\max_{v \in V} c(v)$. Finally, the algorithm returns the set with larger coverage (Algorithm 4 Line 10).

Notice that, when a node is selected, we need to update the coverage and the benefit-cost ratio of nodes gets affected. This may cause unnecessarily high computational costs and the cost will depend on the number of seed nodes selected, which might be very large. To overcome such a deficiency, for threshold greedy, we maintain a list for each threshold, i.e, d_{max}, $d_{max} \cdot (1-\xi)$, $d_{max} \cdot (1-\xi)^2$, , $\cdots d_{max} \cdot (1-\xi)^i$, $\cdots$, such that list i includes a node whose benefit-cost ratio falls in $(d_{max} \cdot (1-\xi)^i, d_{max} \cdot (1-\xi)^{i-1}]$. We further maintain an inverted index for each node v such that each list includes the RR sets covered by v. When a node u is selected as the seed set, it first retrieves all the RR sets covered by u, and then for each RR set R, it updates the benefit-cost ratio of a node $x \in R$ and updates its threshold list if necessary. This can be done all in $O(1)$ cost. Based on the above analysis, we have the following lemma to control the running cost of the budgeted threshold-greedy algorithm.

```
Algorithm 4: Budgeted Threshold-Greedy
Input: Graph $G = (V, E)$, budget $B$, a threshold $\xi$,
       and a set $\mathcal{R}$ of RR sets,
Output: A seed set $S$
1  $S \leftarrow \emptyset$;
2  $d_{max} \leftarrow \max_{v \in V} \Lambda(\{v\})/c(v)$;
3  $d_{min} \leftarrow 1/\max_{v \in V} c(v)$;
4  for $w = d_{max}; w \geq d_{min} \cdot (1-\xi); w = w \cdot (1-\xi)$ do
5      foreach $v \in V$ do
6          if $c(S \cup \{v\}) \leq B$ and $\Lambda(v|S)/c(v) \geq w$ then
7              $S = S \cup \{v\}$;
8          if $c(S) + \min_{v \in V} c(v) > B$ then
9              break;
10 $s \leftarrow \arg\max_{v \in V, c(v) \leq B} \Lambda(\{v\})$;
11 return $\arg\max(\Lambda(\{s\}), \Lambda(S))$;
```

Lemma 5. *Given an input $\mathcal{R}$ of random RR sets, the time complexity of Algorithm 4 can be bounded by $\sum_{R \in \mathcal{R}} |R|$.*

Approximation guarantee. Next, we prove that the budgeted threshold-greedy algorithm still provides an approximate solution with a strong theoretical guarantee. For budgeted threshold-greedy algorithm, we have the following lemma:

Lemma 6. *Let S_i be the set of first i nodes selected by Algorithm 4 Lines 2-9 without skipping any nodes. Let v_i be the i-th node selected. We have that:*

$$\Lambda(S_i) \geq [1 - \prod_{k=1}^{i}(1 - \frac{c(v_k)(1-\xi)}{B})]\Lambda(S'_{opt})$$

Proof. Let S'_{opt} be the optimal solution for maximizing the coverage on $\mathcal{R}$ with budget B.

Let u be the next chosen node to be added to S and w be the threshold value in an iteration. Then we have the following inequalities:

$$\frac{\Lambda(x|S)}{c(x)} = \begin{cases} \geq w, & \textbf{if } x = u \\ \leq w/(1-\xi), & \textbf{if } x \in S'_{opt} \setminus (S \cup \{u\}) \end{cases}$$

To explain, if a node x is not added to S yet and it is not u, then its marginal gain will be no larger than $w/(1-\xi)$, which otherwise should have been added to S in previous iterations. Then the following inequality holds:

$$\frac{\Lambda(u|S)}{c(u)} \geq (1-\xi) \max_{x \in S'_{opt} \setminus S} \frac{\Lambda(x|S)}{c(x)} \tag{2}$$

Let $r_{max} = \max_{x \in S'_{opt} \setminus S} \frac{\Lambda(x|S)}{c(x)}$. Then for any node $x \in S'_{opt} \setminus S$, we have that

$$\Lambda(x|S) \leq c(x) \cdot r_{max}$$

Therefore, we can derive that:

$$\begin{aligned} r_{max} = \max_{x \in S'_{opt} \setminus S} \frac{\Lambda(x|S)}{c(x)} &= \frac{\sum_{x \in S'_{opt} \setminus S} c(x) \cdot r_{max}}{\sum_{x \in S'_{opt} \setminus S} c(x)} \\ \geq \frac{\sum_{x \in S'_{opt} \setminus S} \Lambda(x|S)}{\sum_{x \in S'_{opt} \setminus S} c(x)} &\geq \frac{\sum_{x \in S'_{opt} \setminus S} \Lambda(x|S)}{B} \end{aligned} \tag{3}$$

Combining Equations 2 and 3, we get:

$$\frac{\Lambda(u|S)}{c(u)} \geq (1-\xi) \frac{\sum_{x \in S'_{opt} \setminus S} \Lambda(x|S)}{B} \tag{4}$$

Let S_i be the set of first i nodes selected by Algorithm 4 Lines 2-9. Let v_i be the i-th node selected. By Equation 4, we can derive that:

$$\Lambda(S_i) - \Lambda(S_{i-1}) \geq \frac{c(v_i)(1-\xi)}{B}(\Lambda(S'_{opt}) - \Lambda(S_{i-1}))$$

Given above equation, we can prove that:

$$\Lambda(S_i) \geq [1 - \prod_{k=1}^{i}(1 - \frac{c(v_k)(1-\xi)}{B})]\Lambda(S'_{opt}),$$

using similar technique in the proof of Lemma 3. We omit the details for simplicity. □

The following theorem establishes the approximation guarantee of the budgeted threshold-greedy algorithm.

THEOREM 2. *The budgeted threshold-greedy returns a* $(1 - 1/e^{\eta(1-\xi)})$*-approximate solution for maximizing the coverage on* $\mathcal{R}$ *under budget* B*, where* $(1-\eta)(1-1/e^{1-\xi}) = 1 - \frac{1}{e^{\eta(1-\xi)}}$.

According to Theorem 2, we can see that the approximation ratio is affected by the threshold-greedy parameter ξ. Figure 2 shows the impact of ξ to the worst-case approximation guarantee. As we can observe, when we increase ξ, the worst-case approximation ratio decreases. In the meantime, a larger ξ indicates a faster running time since the budgeted threshold-greedy algorithm can finish with much fewer iterations. Hence, there is a trade-off between the query efficiency and the approximation guarantee. We will examine its impact in our experiment.

Tightened upper bounds. We can apply a similar idea in Section 3.2 to derive a tighter upper bound for the budgeted threshold-greedy algorithm. Recap that we maintain threshold lists such that the i-th list contains the nodes whose benefit-cost ratio falls in $(d_{max} \cdot (1-\xi)^i, d_{max} \cdot (1-\xi)^{i-1}]$. We further maintain the total cost c_i for the nodes in the i-th threshold list. When deriving the upper bound, we repeatedly obtain the nodes with the largest benefit-cost ratio until the cost exceeds B. A naive solution is to first sort all the nodes according to their benefit-cost ratio. Yet, since we have maintained the threshold lists, we check the total cost c_1 of the first threshold list and add all the nodes to $\kappa(S, B)$ if the total cost is no larger than B. Then we further check if $c_2 < B - c_1$, and if it is true, we add all the nodes in the second threshold list to $\kappa(S, B)$ until the current i-th threshold list incurs a cost larger than $B - \sum_{j=1}^{i-1} c_j$. Then, we sort the nodes in the i-th threshold list according to their benefit-cost ratio and repeatedly add the nodes with the largest benefit-cost ratio in this list until adding a new node makes the total budget exceed B. We add the last node as a broken node and the seed set returns as $\kappa(S, B)$. But this may make the total running cost for bound refinement linearly depend on the seed set size. To alleviate such a dependency, we impose that the time for deriving upper bounds does not exceed the time for random RR set generation.

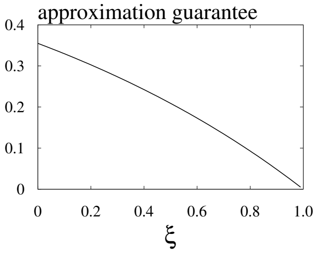


Figure 2: The impact of ξ to worst case approximation guarantee of budgeted threshold-greedy.

4. TIME COMPLEXITY ANALYSIS

In this section, we present theoretical analysis for our proposed IMAGE framework. Notice that IMAGE indicates the full-fledged algorithm mentioned in Section 3.3. We present the detailed analysis for IMAGE and omit the result for IMAGE-BR for simplicity. We first define several constants that only depend on the input graph and budget B. Let S^- be a set such that adding any node from $V \setminus S^-$ will make the total budget exceed B. Further define $\mathcal{S}^-$ as the set that includes all such S^- and let $\tau_B = |\mathcal{S}^-|$. Now assume that we generate a set S_c by repeatedly adding the node with the smallest cost not in S_c until adding another node will exceed the budget. Let k_{max} be the number of nodes picked in S_c. Then, it is easy to verify that $n^{k_{max}}$ is an upper bound of τ_B. Besides, k_{max} is a lower bound of the expected influence under budget B. We first have the following lemma to control the number of RR set required by IMAGE.

LEMMA 7. *Let* ϵ *be an approximation error parameter and* δ *be a failure probability. Let* $\mathcal{R}$ *be the set of RR sets sampled by IMAGE and* $z = 1/e^{\eta(1-\xi)}$*. IMAGE provides a* $(1-z-\epsilon)$*-approximate solution with* $1-\delta$ *probability when*

$$|\mathcal{R}| \geq \frac{2n \cdot \left((1-z)\sqrt{\ln\frac{2}{\delta}} + \sqrt{(1-z)(k_{max}\ln n + \ln\frac{2}{\delta})}\right)^2}{\epsilon^2 k_{max}}.$$

Based on Lemma 7, we set θ_0 and θ_{max} for IMAGE as follows to provide approximation guarantee. We set

$$\theta_{max} = \frac{2n \cdot \left((1-z)\sqrt{\ln\frac{6}{\delta}} + \sqrt{(1-z)(k_{max}\ln n + \ln\frac{6}{\delta})}\right)^2}{\epsilon^2 k_{max}}, \tag{5}$$

which provides $(1-z-\epsilon)$-approximate solution with $1 - \delta/3$ probability when we sample θ_{max} random RR sets. Let k_{min} be the number of nodes selected by repeatedly selecting nodes with the maximum cost until no node can be added. Then, we further set:

Table 2: Dataset Statistics ($M = 10^6, B = 10^9$)

Name	n	m	Type	Avg. deg
Pokec	1.6M	30.6M	directed	37.5
Orkut	3.1M	117.2M	undirected	76.3
Weibo	1.8M	414M	directed	462.7
Twitter	41.7M	1.5B	directed	70.5

$$\theta_0 = \frac{2n \cdot \left((1-z)\sqrt{\ln\frac{6}{\delta}} + \sqrt{(1-z)(k_{min}\ln n + \ln\frac{6}{\delta})}\right)^2}{\epsilon^2 n} \tag{6}$$

in which case we have the minimum size of seed set and the expected influence is n, and the above number θ_0 is the minimum number of RR sets required by such best scenario. By setting θ_{max} and θ_0 according to Equations 5 and 6 respectively, we have the following lemma.

Lemma 8. *When IMAGE terminates, it provides a $(1 - 1/e^{\eta(1-\xi)} - \epsilon)$-approximate solution with $1 - \delta$ probability.*

Finally, we have the following theorem to bound the expected running time of IMAGE algorithm.

Theorem 3. *Under both IC and LT model, IMAGE runs in $O((\ln(1/\tau_B) + \ln(1/\delta))(n + m)\epsilon^{-2})$ expected time.*

5. EXPERIMENTS

This section experimentally evaluates our solutions against alternatives. All experiments are conducted on a Linux machine with an Intel Xeon 2.70GHz CPU and 400GB memory.

5.1 Experimental Setup

Datasets. We test on four real social networks: Pokec [29], Orkut [29], Weibo [38], and Twitter [26], which are widely used in the literature for evaluating influence maximization algorithms. Table 2 shows the statistics of each dataset.

Algorithms. We evaluate our IMAGE-BR and IMAGE algorithms against the baseline solution [31], dubbed as *Baseline*, which applies the budgeted greedy on random RR sets with no bound refinement. The IMAGE-BR algorithm includes the bound refinement technique mentioned in Section 3.2. The IMAGE algorithm is the full-fledged algorithm mentioned in Section 3.3. We implement all algorithms in C++ and compile them with full optimization. For each algorithm, we repeat 5 times and report the average as the evaluation result when evaluating the running time, expected influence, and number of RR sets.

Parameters. We evaluate our algorithms on both IC model and LT model. Following existing work [37, 36, 35], for both IC model and LT model, we set the probability of each edge $\langle u, v\rangle$ as $1/d_{in}(u)$, where $d_{in}(u)$ is the in-degree of node u.

Based on the report on news and social media [1, 2, 3, 4, 5, 6], only the top $0.001\% - 0.0001\%$ users could be regarded as social media influencers. We set the top 0.001% of the user as the influencers. According to the statistics of pay-rate to social media influencers and non-influencers [1, 2, 3, 4, 5, 6], the payment to influencers is assigned as $c(u) = 0.01 \cdot d_{in}(u)$ and the pay-rate to non-influencers is set as $c(u) = 2 \cdot d_{in}(u)$. This is consistent with real scenarios [1, 2, 3, 4, 5, 6] where the pay-rate per follower for celebrities are usually much lower than the pay-rate per follower for normal users. For budget B, we set the default value to be $0.0002 \cdot n$. Recall that our IMAGE includes a parameter ξ to balance the cost of node selection and the approximation guarantee. We tune the parameter in Section 5.4. We observe that when $\xi = 0.05$ achieves a good trade-off in all tested datasets and set it as the default value in all remaining experiments.

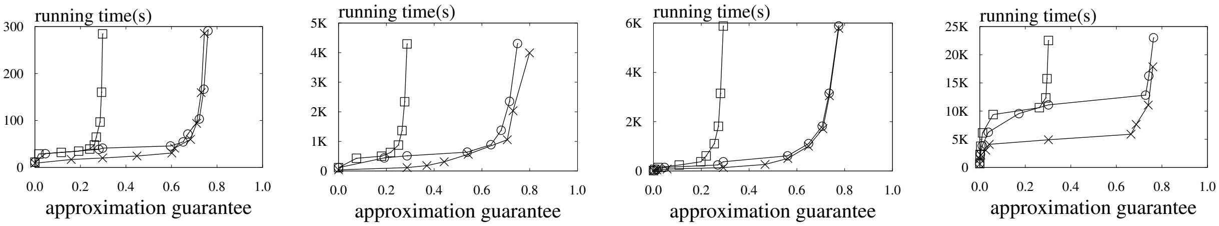
5.2 Efficiency and Effectiveness

In the first set of experiments, we evaluate the running time of our methods against the baseline methods. We examine the running time required to achieve a user-defined approximation ratio in the range of $[0, 0.9)$. Note that for a user-defined approximation ratio x that is no larger than the worst-case guarantee y, ϵ is set as $y - x$. However, Baseline cannot provide an answer when the user-defined approximation ratio is larger than its worst-case guarantee. In contrast, our IMAGE-BR and IMAGE can still provide approximation ratio that is larger than the worst-case ratio since the upper bound we derived are typically far better than the worst-case bound.

Figure 3 reports the running time under the IC model. As we can observe, our IMAGE-BR and IMAGE are more efficient than Baseline to provide the same approximation guarantee. For example, when the approximation ratio is 0.3, IMAGE-BR (resp. IMAGE) is up to 15x (resp. 40x) faster than Baseline. Our IMAGE is further up to 4x faster than IMAGE-BR. To explain, IMAGE uses the budgeted threshold greedy and it can help reduce the cost for bound estimation and seed selection, achieving smaller running costs. Figure 4 reports the results under the LT model. It shows a similar trend: IMAGE is still 1 order of magnitude faster than Baseline and IMAGE-BR is several times faster than Baseline when the approximation ratio is 0.3.

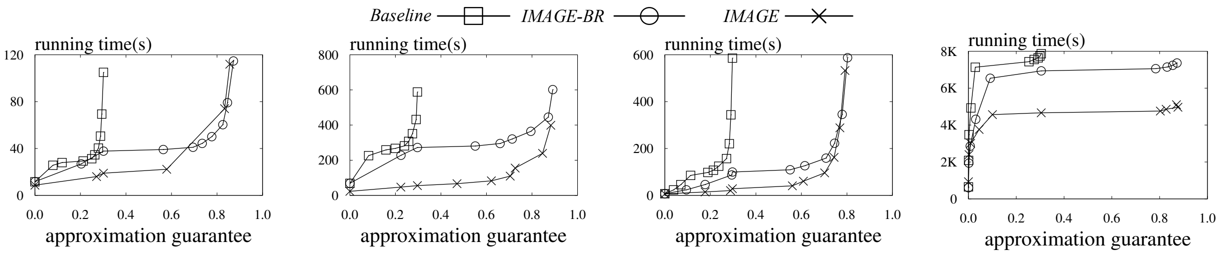
In the next set of experiments, we examine the impact of the number of RR sets to the approximation ratio of each algorithm. For the ease of comparison, we fix the initial number of random RR sets to be $2^4 \times 10^3$. By this initial set, we can examine the approximation ratio achieved by each method with the same number of random RR sets. Figure 5 (resp. Figure 6) reports the approximation ratio when the number of RR sets varies under the IC model (resp. LT model). As we can observe, in both IC and LT model, IMAGE and IMAGE-BR achieve better approximation ratio than Baseline since IMAGE and IMAGE-BR adopt the new bound estimation method. In the meantime, with the same number of RR sets, IMAGE and IMAGE-BR achieves identical approximation guarantee in most of the cases and their approximation ratios increase with the number of RR sets. We note that IMAGE may require a slightly larger number of RR sets to achieve the same approximation ratio in some cases. However, as shown in Figures 3-4, IMAGE is still more efficient than the IMAGE-BR to achieve the same approximation ratio. To explain, IMAGE adopts the budgeted threshold greedy, which makes it more efficient in bound estimation and node selection, thus saving the running costs and outperforming IMAGE-BR.

Finally, we examine the quality of the seed set returned by each algorithm. As shown in Figures 7-8, the expected influence of the seed set returned by the Baseline, IMAGE-BR, and IMAGE are identical on both the IC and LT models. When the same number of RR sets are sampled, all the three algorithms tend to provide seed set with identical quality. Moreover, with a larger number of RR sets, all



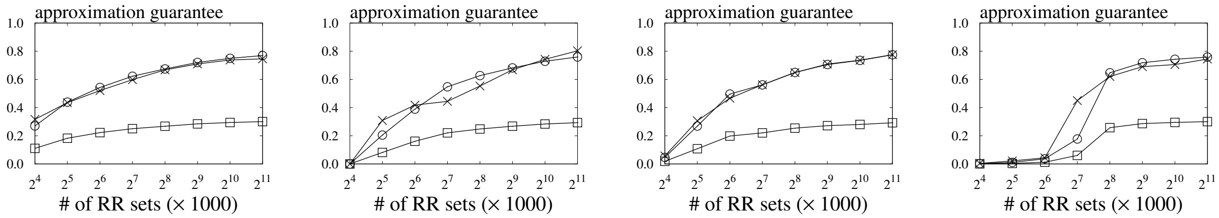
(a) Pokec (b) Orkut (c) Weibo (d) Twitter

Figure 3: Running time vs. approximation guarantee under the IC model



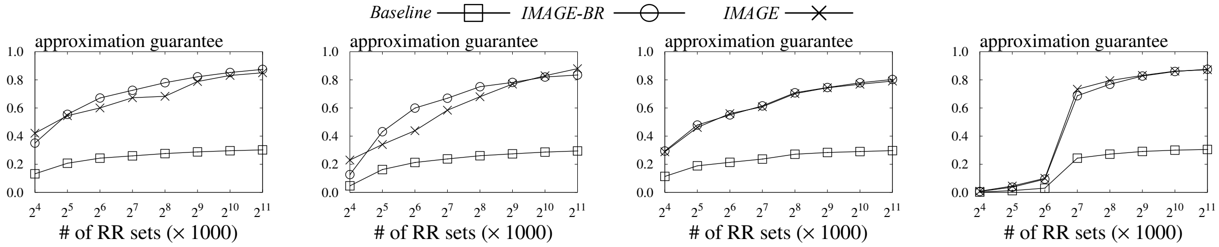
(a) Pokec (b) Orkut (c) Weibo (d) Twitter

Figure 4: Running time vs. approximation guarantee under the LT model



(a) Pokec (b) Orkut (c) Weibo (d) Twitter

Figure 5: Approximation guarantee vs. # of RR sets under the IC model



(a) Pokec (b) Orkut (c) Weibo (d) Twitter

Figure 6: Approximation guarantee vs. # of RR sets under the LT model

three algorithms tend to provide seed with better quality. This is expected since with more RR sets, the estimation of the expected influence becomes more accurate and helps the greedy algorithms select node sets with better quality. However, since Baseline only uses the worst-case bound to derive the approximation ratio, it cannot stop promptly when it has actually selected a seed set satisfying the user-specified approximation ratio, thus wasting the computational costs.

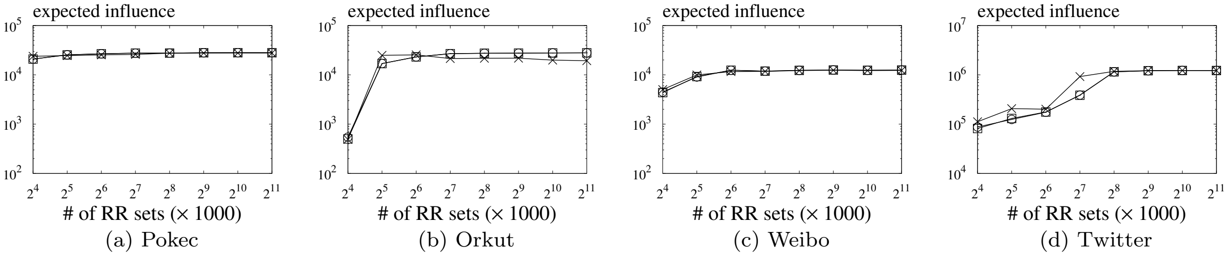
In summary, our IMAGE is orders of magnitude faster than alternatives while providing similarly high-quality seed set, which is the preferred choice for the BIM problem.

5.3 Cost of Bound Estimation and Greedy

Next, we examine the cost of three major phases: the RR set generation, bound estimation, and node selection. Figures 9 and 10 reports the running cost of each phase on the four datasets when the approximation ratio is 0.3. Notice that PK, WB, OR, and TW are short for Pokec, Weibo, Orkut, and Twitter, respectively. Since Baseline and IMAGE-BR adopt the same greedy algorithm (Algorithm 2), their running cost with greedy is similarly high. In contrast, IMAGE adopts the budgeted threshold greedy algorithm (Algorithm 4) for the node selection, and may select multiple nodes in a single iteration, thus reducing the running costs. Besides, as we can observe, the bound estimation cost of IMAGE-BR increases when the size of the graph increases, and may dominate the cost of RR set generation, e.g., on Twitter dataset under LT model. However, the bound estimation cost of IMAGE is far less than IMAGE-BR. To explain, as mentioned in Section 3.3, we can use the threshold list maintained in Algorithm 4 to reduce the computational costs in the bound estimation phase.

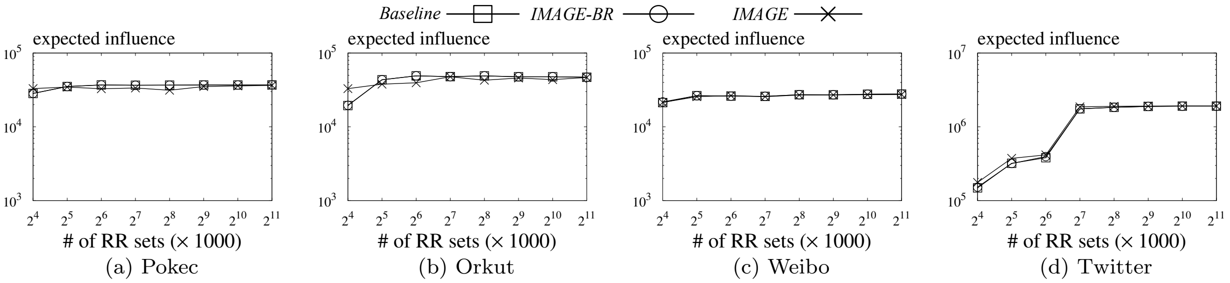
5.4 Impact of ξ

Recap that our IMAGE algorithm includes a parameter ξ that provides a trade-off between the query efficiency and approximation guarantee. We inspect the impact of ξ to IMAGE and report the results on two representative datasets:



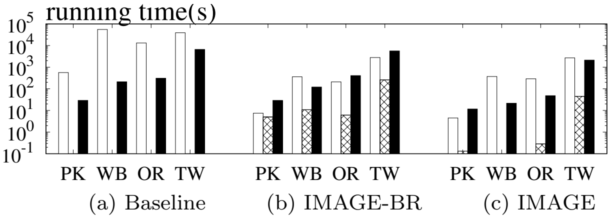
(a) Pokec (b) Orkut (c) Weibo (d) Twitter

Figure 7: Expected influence vs. # of RR sets under the IC model



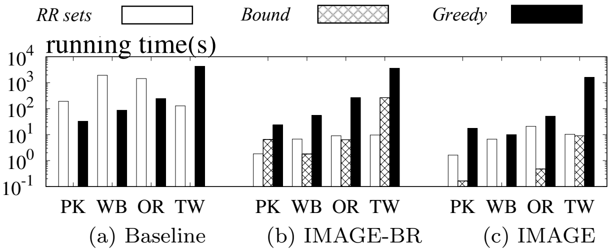
(a) Pokec (b) Orkut (c) Weibo (d) Twitter

Figure 8: Expected influence vs. # of RR sets under the LT model



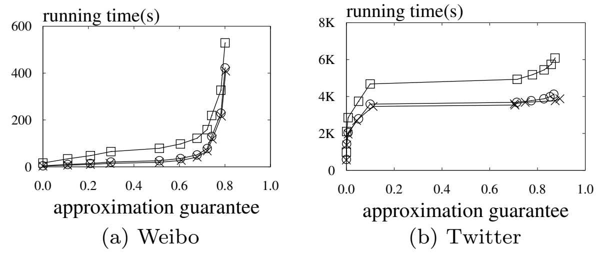
(a) Baseline (b) IMAGE-BR (c) IMAGE

Figure 9: Cost of different phases: IC model



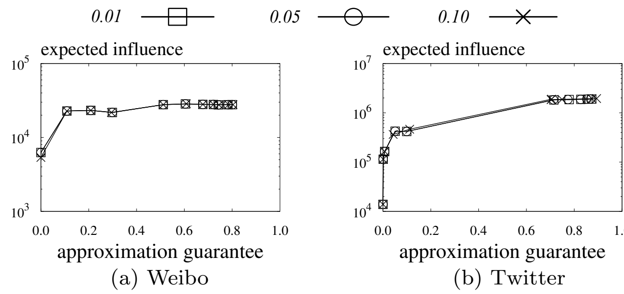
(a) Baseline (b) IMAGE-BR (c) IMAGE

Figure 10: Cost of different phases: LT model



(a) Weibo (b) Twitter

Figure 11: ξ vs. running time: LT model



(a) Weibo (b) Twitter

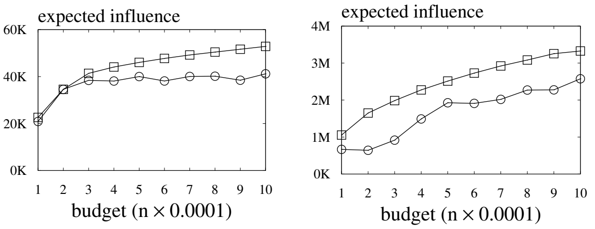
Figure 12: ξ vs. expected influence: LT model

Weibo and Twitter under the LT model. As shown in Figures 11, a smaller ξ indicates a larger running cost, e.g. with $\xi = 0.01$ on Twitter dataset. To explain, when ξ is quite small, it is highly likely that the algorithm involves many iterations to select the seed nodes, which increases the running cost. For $\xi = 0.05$ and $\xi = 0.1$, they have the similar running time and expected influence as shown in Figures 11-12. However, when ξ increases, it indicates a looser approximation guarantee and may affect the quality of the selected seed set in worst-case scenarios. We have also tested the impact of ξ under the IC model and have a similar observation. For the interest of space, we omit the results here. Interested readers may refer to our technical report [7] for the details.

According to above experimental study, we set $\xi = 0.05$ as the default setting since it strikes a good balance between the running cost and worst-case approximation guarantee.

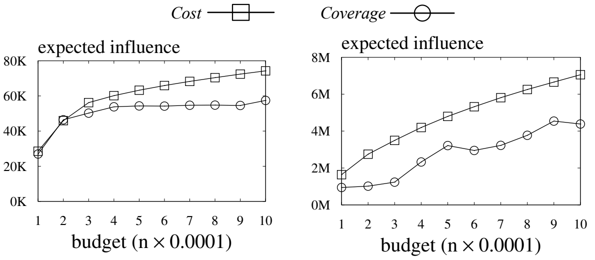
5.5 BIM v.s. IM

Finally, we examine the effectiveness of BIM against classic IM when we consider cost differences. We first apply the classic IM algorithms to solve the BIM problem and choose the seed set with the maximum coverage greedy algorithm (Algorithm 1) select the first the nodes and skip a node if adding it will make the cost exceed the budget, until no node can be added into the seed set. We compare the expected influence of the seed set returned by using classic IM algorithm and the seed set returned by using our BIM algorithm IMAGE. As we can see in Figures 13-14, the solution considering cost achieves far better expected influence when we provide different settings of the budget. In the meantime, for classic IM algorithms, an increased budget does not always guarantee to return a seed set with an increased expected influence. To explain, with the growth of the budget, the classic IM algorithms may select some nodes with a large expected influence but with a low benefit-cost ratio, resulting in inferior seed sets compared to the seed sets returned with even smaller budgets. In contrast, for BIM algorithms,



(a) Pokec (b) Twitter

Figure 13: IM vs. BIM under IC model



(a) Pokec (b) Twitter

Figure 14: IM vs. BIM under LT model

we can see that with the growth of the budget, BIM algorithms always return seed sets with an increased expected influence since BIM algorithms take the cost of each node into consideration as well. This indicates that BIM is the preferred solution when different users have different costs.

6. RELATED WORK

Kempe et al. [24] present the first study on influence maximization problem, and prove that the influence maximization problem is NP-hard. They provide a greedy framework and show that the proposed algorithm obtains a $(1-1/e-\epsilon)$-approximate solution for both IC and LT models. However, the time complexity of the proposed algorithm is $\Omega(kmn \cdot poly(1/\epsilon))$, which is too expensive for large social networks. After that, a large number of research works [8, 10, 12, 13, 14, 15, 16, 17, 19, 20, 21, 22, 27, 34] aim to improve Kempe et al.'s solution to make their algorithms more efficient and scalable to large social networks. Many of them focus on heuristic solutions, which tend to improve the practical performance over previous solutions. However, such solutions provide no guarantee on the quality of the returned result.

To tackle such a challenging issue, Borgs et al. present a seminar work [10] and propose the random reverse reachable (RR) set technique to solve the influence maximization problem. The proposed solution reduces the time complexity to almost linear to the graph size. In particular, the algorithm can return a $(1-1/e-\epsilon)$-approximate solution with $1-1/n$ probability with $O(k(m+n)\epsilon^{-3}\log^2 n)$ running time. Then, a plethora of research works [37, 36, 32, 35] improve the efficiency on IM problem based on the random RR set based techniques. Tang et al. [37] propose TIM to reduce the number of random RR samples, and show that the time complexity can be improved to $O(k(m+n)\epsilon^{-2}\log n)$ while providing the same approximation guarantee with the same success probability under both LT and IC model. Tang et al. [36] further present IMM, an enhanced version of TIM, by exploring the martingale property of the random RR sets and can reuse some random samples even though there are some weak dependencies between different random RR sets. They show that IMM retains the same time complexity and approximation guarantee as TIM but is far more efficient in practice since it reduces the number of random RR sets. Nguyen et al. [32, 33] propose SSA, SSA-Fix, D-SSA and D-SSA-Fix to further improve the practical performance over TIM and IMM. The main idea is to reduce the dependency to the seed set size by a verification phase to see if the selected seed is good or not, thus saving running costs. Lately, Tang et al. [35] propose OPIM-C to derive tighter upper and lower bounds so that the algorithm can terminate as soon as the approximation ratio is satisfied, reducing the running costs. However, all these solutions focus on classic IM problem and do not take into account the cost differences.

For the BIM problem, a line of research works focus on developing heuristic algorithms, e.g., [30, 23, 28], to reduce the computational costs. Such heuristics, however, provides no guarantee on the returned answer. Khuller et al. [25] propose the budgeted greedy so that the algorithm returns a $(\frac{1-1/e}{2})$-approximate solution if one can calculate the exact expected influence of any set. They further claim that the approximation ratio can be improved to $1-1/\sqrt{e}$. However, as shown in [39], the proof contains loopholes and the approximation ratio does not hold. Nguyen et al. [31] extend the RR set based solution to the budgeted IM problem by first sampling a sufficient number of RR set and then applying the budgeted greedy algorithm. They claim that the proposed algorithm returns a $(1-1/\sqrt{e}-\epsilon)$-approximate solution, while should be $(\frac{1-1/e}{2}-\epsilon)$ since their proof contains the same loopholes as the one in [25]. As we have shown in our experiment, the proposed solution is still inefficient and leaves much room for improvement, which motivates us to propose the IMAGE framework.

7. CONCLUSIONS

In this paper, we present an efficient framework IMAGE for BIM problem. We propose new bound refinement strategies and node selection optimizations to improve the performance for such queries. We further present theoretical analysis and show that our proposed solution has a time complexity of $O((\ln 1/\delta + \ln \tau_B)(n+m)/\epsilon^2)$. Experiments show that our solution is far more efficient than alternatives.

8. ACKNOWLEDGMENTS

Sibo Wang is supported by Hong Kong RGC ECS Grant No. 24203419, CUHK Direct Grant No. 4055114, NSFC Grant No. U1936205, and a CUHK University Startup Grant. Jeffrey Xu Yu is supported by Hong Kong RGC Grant No. 14203618 and No. 14202919.

9. APPENDIX

We omit some of the proofs due to the interest of space. Please refer to [7] for omitted proofs.

Proof of Lemma 3. Before proving Lemma 3, we first prove the following lemma.

LEMMA 9. *Given an input set $\mathcal{R}$ of random RR sets and budget B, let v_i be the node selected in the i-th iteration and S_i be the set of nodes selected after iteration i ($i = 1, 2, 3, \cdots$) by the budgeted greedy algorithm (Algorithm 2 Lines 3-6) based on $\mathcal{R}$. The following inequality holds:*

$$\Lambda(S_i) - \Lambda(S_{i-1}) \geq \frac{c(v_i)}{B} \cdot (\Lambda(S_{opt}) - \Lambda(S_{i-1})), \qquad (7)$$

where S'_{opt} is the optimal set that maximizes the coverage on $\mathcal{R}$ under budget B. □

It is obvious that $\Lambda(S'_{opt}) - \Lambda(S_{i-1})$ is smaller than $\Lambda(S'_{opt} \setminus S_{i-1})$. For each seed node v in $S'_{opt} \setminus S_{i-1}$, the ratio of the marginal gain $\Lambda(v|S_{i-1})$ on $\mathcal{R}_1$ to its cost $c(v)$ is at most $(\Lambda(S_i) - \Lambda(S_{i-1}))/c(v_i)$. Due to the property of the budgeted greedy algorithm, v_i maximizes the ratio over all candidate seed nodes which are not selected into S_{i-1}. Because the total cost of seed set is constrained by budget B, then $\Lambda(S'_{opt} \setminus S_{i-1})$ is at most $B \cdot (\Lambda(S_i) - \Lambda(S_{i-1}))/c(v_i)$. Therefore, we could get the following equation:

$$\Lambda(S'_{opt}) - \Lambda(S_{i-1}) \leq B \cdot \frac{\Lambda(S_i) - \Lambda(S_{i-1})}{c(v_i)}$$

Lemma 9 has been proved. Next, we prove Lemma 3. It is known that $\forall i, \Lambda(S_1)/c_1 \geq \Lambda(v_i)/c_i$. Because the greedy algorithm chooses the most cost-effective node first. We assume that the lemma holds for iterations $1, 2, \ldots, i-1$. Then, at the i^{th} iteration we could conclude that:

$$\begin{aligned}\Lambda(S_i) &= \Lambda(S_{i-1}) + [\Lambda(S_i) - \Lambda(S_{i-1})] \\ &\geq \Lambda(S_{i-1}) + \frac{c(v_i)}{B} \cdot (\Lambda(S'_{opt}) - \Lambda(S_{i-1})) \\ &= (1 - \frac{c(v_i)}{B}) \cdot \Lambda(S_{i-1}) + \frac{c(v_i)}{B} \cdot \Lambda(S'_{opt}) \\ &\geq (1 - \frac{c(v_i)}{B}) \cdot (1 - \prod_{k=1}^{i-1}(1 - \frac{c(v_k)}{B})) \cdot \Lambda(S'_{opt}) \\ &+ \frac{c(v_i)}{B} \cdot \Lambda(S'_{opt}) = \left(1 - \prod_{k=1}^{i}(1 - \frac{c(v_k)}{B})\right) \cdot \Lambda(S'_{opt})\end{aligned}$$

This finishes the proof of Lemma 3. □

Proof of Theorem 1. We omit the proof since it can follow the proof of Theorem 2 by setting $\xi \to 0$ and $\eta = \beta$. □

Proof of Theorem 2. First, we introduce a classic inequality. If we have $a_1, \cdots, a_n \in R^+$ which satisfies that $\sum_{i=1}^n a_i = \alpha A$, the objective function

$$(1 - \prod_{i=1}^{n}(1 - \frac{a_i}{A})) \quad (8)$$

achieves its minimum value: $1 - (1 - \alpha/n)^n$, when $a_1 = a_2 = \cdots = a_n = \alpha A/n$, for $A, \alpha > 0$. Based on Lemma 3, we can prove Theorem 2 by case analysis. We first assume that we have eliminated any node whose budget exceeds B. Therefore, any nodes remained have a cost no larger than B. Then, we have two cases.

- **Case 1**: There exists a node whose coverage is larger than $\eta\Lambda(S'_{opt})$, then in Algorithm 4 Line 10, node s selected provides at least η-approximate solution.
- **Case 2**: There does not exist no such a node u whose coverage is no smaller than $\eta \cdot \Lambda(S'_{opt})$. Let S be the set selected by Algorithm 4 Lines 4-9. We have two sub-cases.

– **Case 2.1:** $c(S) < \eta B$, it is not difficult for us to have that $c(u) > (1-\eta) \cdot B$, $\forall u \notin S$. Because if $\exists u \notin S$ and $c(u) \leq (1-\eta) \cdot B$, we can add u to S, which is conflicted with Algorithm 4. We assume that $S \neq S'_{opt}$ (otherwise the approximation ratio is 1) and $\eta \leq \frac{1}{2}$. Then, $S'_{opt} \setminus S$ contains at most one such u, i.e., a node with budget larger than $(1-\eta) \cdot B$. Otherwise, $c(S'_{opt}) > B$. Since Λ is submodular, we have:

$$\begin{aligned}\Lambda(S'_{opt}) &= \Lambda\left((S'_{opt} \cap S) \cup (S'_{opt} \setminus S)\right) \\ &\leq \Lambda(S'_{opt} \cap S) + \Lambda(S'_{opt} \setminus S) \leq \Lambda(S) + \Lambda(\{u\})\end{aligned}$$

According to Case 2, we have $\Lambda(\{u\}) < \eta\Lambda(S'_{opt})$. Therefore, $\Lambda(S) \geq (1-\eta)\Lambda(S'_{opt})$.

– **Case 2.2:** $c(S) \geq \eta B$. According to Lemma 6 and the inequality for Equation 8, we have that:

$$\begin{aligned}\Lambda(S_i) &\geq [1 - \prod_{k=1}^{i}(1 - \frac{c(v_k)(1-\xi)}{B})] \cdot \Lambda(S'_{opt}) \\ &\geq [1 - (1 - \frac{\eta(1-\xi)}{i})^i] \cdot \Lambda(S'_{opt}) \quad (9)\end{aligned}$$

Note that $\Lambda(S) \geq \Lambda(S_i) \geq (1 - 1/e^{\eta(1-\xi)}) \cdot \Lambda(S'_{opt})$ holds if $c(S_i) \geq \eta B$. If $c(S_i) < \eta B$, we analyze case by case.

∗ **Case 2.2.i:** $c(S_i) < \eta B$ while $v_{i+1} \in S'_{opt}$ and it could not be added to S_i. Then we could construct another seed set $S^+ = S'_{opt} \setminus \{v_{i+1}\}$. According to Case 2.1, we get that $\Lambda(S^+) \geq (1-\eta)\Lambda(S'_{opt})$. Let S^+_{opt} be the optimal seed set with the maximum coverage on $\mathcal{R}$ under a budget of $c(S^+)$. According to the definition, we know $\Lambda(S^+_{opt}) \geq \Lambda(S^+)$. Besides, $c(S) \leq B$ and $c(S^+) = c(S^+_{opt}) \leq \eta B$. We then construct another set S^f_j as follows: we add $v_1, v_2, v_3, \cdots, v_{j-1}$ (v_x is the x-th node selected by Algorithm 4 without skipping any node) to S^f_j until adding v_j will cause the total budget to exceed $c(S^+)$. We then add a fraction of node v_j to S^f_j so that the total budget is exactly $c(S^+)$. If this node v_j can be added fully to S_i, then, since we only add part of this node to S^f_j, we can easily conclude that $\Lambda(S_i) \geq \Lambda(S^f_j)$. If this node v_j can not be added into S_i, which means $c(v_j) \geq (1-\eta)B$ and $\eta \leq \frac{1}{2}$, then this node v_j will not be added to S^f_j as well. To explain, the cost of v_j exceeds budget $c(S^+)$, and by removing such a node, it does not affect the node selection and the optimal solution. Therefore, we simply discard this node v_j and in both cases $\Lambda(S_i) \geq \Lambda(S^f_j)$. For S^f_j, it can be verified that $\Lambda(S^f_j) \geq (1 - 1/e^{1-\xi})\Lambda(S^+_{opt})$ with Equation 9 since $B = c(S^f_j) = c(S^+_{opt})$ and $\eta = 1$. Then, in Case 2.2.i, we have that:

$$\begin{aligned}&\Lambda(S) \geq \Lambda(S_i) \geq \Lambda(S^f_j) \geq (1 - 1/e^{1-\xi}) \cdot \Lambda(S^+_{opt}) \\ &\geq (1 - 1/e^{1-\xi}) \cdot \Lambda(S^+) \geq (1-\eta)(1 - 1/e^{1-\xi}) \cdot \Lambda(S'_{opt})\end{aligned}$$

∗ **Case 2.2.ii:** $c(S_i) < \eta B$ and $c(v_{i+1} \cup S_i) > B$ but $v_{i+1} \notin S'_{opt}$. In this case, it is safe to skip node u_{i+1} since Equation 7 still holds for any node selected next (if any). Hence, the inequality for Equation 8 still holds. Then, Case 2.2.ii either falls to Case 2.2.i or makes the budget no smaller than $\eta \cdot B$, where the solution provides a $1 - 1/e^{\eta(1-\xi)}$ approximation ratio.

The above case analysis indicates that we could get the lower bound of $\Lambda(S)$, when $(1-\eta)(1-1/e^{1-\xi}) = 1 - \frac{1}{e^{\eta(1-\xi)}}$. In this case, $\eta < 0.5$, which meets our requirement of η in the discussion of Case 2.1. Let S_{opt} be the seed set maximizing the expected influence under budget B, then $\Lambda(S'_{opt}) \geq \Lambda(S_{opt})$. This finishes the proof. □

10. REFERENCES

[1] https://www.bbc.com/zhongwen/simp/chinese-news-42992794.

[2] https://www.bbc.com/news/technology-50418807.

[3] https://tech.qq.com/a/20190208/002429.htm.

[4] https://xw.qq.com/cmsid/20190803A0B2SZ00.

[5] https://tech.sina.com.cn/i/2019-05-23/doc-ihvhiews4077548.shtml.

[6] https://m.21jingji.com/article/20171205/herald/cb929394ffb50db7df2b7f962bc63d37.html.

[7] https://sites.google.com/site/bimvldb2020tr/.

[8] A. Arora, S. Galhotra, and S. Ranu. Debunking the myths of influence maximization: An in-depth benchmarking study. In *SIGMOD*, pages 651–666, 2017.

[9] A. Badanidiyuru and J. Vondrák. Fast algorithms for maximizing submodular functions. In *SODA*, pages 1497–1514, 2014.

[10] C. Borgs, M. Brautbar, J. T. Chayes, and B. Lucier. Maximizing social influence in nearly optimal time. In *SODA*, pages 946–957, 2014.

[11] C. Budak, D. Agrawal, and A. El Abbadi. Limiting the spread of misinformation in social networks. In *WWW*, pages 665–674, 2011.

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

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

[14] W. Chen, Y. Yuan, and L. Zhang. Scalable influence maximization in social networks under the linear threshold model. In *ICDM*, pages 88–97, 2010.

[15] S. Cheng, H. Shen, J. Huang, W. Chen, and X. Cheng. Imrank: influence maximization via finding self-consistent ranking. In *SIGIR*, pages 475–484, 2014.

[16] S. Cheng, H. Shen, J. Huang, G. Zhang, and X. Cheng. Staticgreedy: solving the scalability-accuracy dilemma in influence maximization. In *CIKM*, pages 509–518, 2013.

[17] E. Cohen, D. Delling, T. Pajor, and R. F. Werneck. Sketch-based influence maximization and computation: Scaling up with guarantees. In *CIKM*, pages 629–638, 2014.

[18] A. Ene and H. L. Nguyen. A nearly-linear time algorithm for submodular maximization with a knapsack constraint. In *ICALP*, pages 53:1–53:12, 2019.

[19] S. Galhotra, A. Arora, and S. Roy. Holistic influence maximization: Combining scalability and efficiency with opinion-aware models. In *SIGMOD*, pages 743–758, 2016.

[20] S. Galhotra, A. Arora, S. Virinchi, and S. Roy. ASIM: A scalable algorithm for influence maximization under the independent cascade model. In *WWW*, pages 35–36, 2015.

[21] A. Goyal, F. Bonchi, and L. V. S. Lakshmanan. A data-based approach to social influence maximization. *PVLDB*, 5(1):73–84, 2011.

[22] A. Goyal, W. Lu, and L. V. S. Lakshmanan. CELF++: optimizing the greedy algorithm for influence maximization in social networks. In *WWW*, pages 47–48, 2011.

[23] S. Han, F. Zhuang, Q. He, and Z. Shi. Balanced seed selection for budgeted influence maximization in social networks. In *PAKDD*, pages 65–77, 2014.

[24] D. Kempe, J. M. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *SIGKDD*, pages 137–146, 2003.

[25] S. Khuller, A. Moss, and J. Naor. The budgeted maximum coverage problem. *Inf. Process. Lett.*, 70(1):39–45, 1999.

[26] H. Kwak, C. Lee, H. Park, and S. B. Moon. What is twitter, a social network or a news media? In *WWW*, pages 591–600, 2010.

[27] J. Lee and C. Chung. A fast approximation for influence maximization in large social networks. In *WWW*, pages 1157–1162, 2014.

[28] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. M. VanBriesen, and N. S. Glance. Cost-effective outbreak detection in networks. In *SIGKDD*, pages 420–429, 2007.

[29] J. Leskovec and A. Krevl. SNAP Datasets: Stanford large network dataset collection. http://snap.stanford.edu/data.

[30] H. Nguyen and R. Zheng. On budgeted influence maximization in social networks. *IEEE J. Sel. Areas Commun.*, 31(6):1084–1094, 2013.

[31] H. T. Nguyen, T. N. Dinh, and M. T. Thai. Cost-aware targeted viral marketing in billion-scale networks. In *INFOCOM*, pages 1–9, 2016.

[32] H. T. Nguyen, M. T. Thai, and T. N. Dinh. Stop-and-stare: Optimal sampling algorithms for viral marketing in billion-scale networks. In *SIGMOD*, pages 695–710, 2016.

[33] H. T. Nguyen, M. T. Thai, and T. N. Dinh. Stop-and-stare: Optimal sampling algorithms for viral marketing in billion-scale networks. *CoRR*, abs/1605.07990, 2016.

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

[35] J. Tang, X. Tang, X. Xiao, and J. Yuan. Online processing algorithms for influence maximization. In *SIGMOD*, pages 991–1005, 2018.

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

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

[38] J. Zhang, B. Liu, J. Tang, T. Chen, and J. Li. Social influence locality for modeling retweeting behaviors. In *IJCAI*, pages 2761–2767, 2013.

[39] P. Zhang, Z. Bao, Y. Li, G. Li, Y. Zhang, and Z. Peng. Trajectory-driven influential billboard placement. In *SIGKDD*, pages 2748–2757, 2018.